

NOVIEMBRE 1987
REVISTA PARA USUARIOS DE LA NORMA
LOAD MSX
AÑO 2 Nº 19
A 6 REP.ARGENTINA
COMO COMANDAR LA MSX
NUEVO SOFT:
• PARA HACER CONTABILIDAD
• PARA MANEJAR LA MEMORIA
LA POTENCIA DE LA MSX2
CONCURSO ULTIMO MES

EN INFORMATICA TALENT MSX HACE ESCUELA.

Y una prueba de ello, son algunos de los establecimientos que han incorporado computadoras Talent MSX como herramienta de apoyo pedagógico.

CAPITAL FEDERAL:

COLEGIO DR. ANGEL GALLARDO
ESC. Nº 10 NTRA. SRA. DEL CARMEN
ASOC. CRISTIANA DE JOVENES
INST. DE APREND. HOLISTICO
COLEGIO JESUS MARIA
ESC. Nº 2 "D. F. SARMIENTO"
COLEGIO JESUS MARIA
ESC. MENTES SIN FRONTERA
UNIVERSIDAD DEL SALVADOR
E.N.E.T. Nº 19 "ALEJANDRO VOLTA"
CTRO. DE EST. DE INF. CIPA
C.O.D.I.C.E.
CENEA
C.L.A.I.C.E.
PROG. CULT. EN SINDICATOS
SUBSECRETARIA DE INF. Y DES.
FUND. NTRA. SRA. DE LA MERCED
INST. MARTIN M. DE GÜEMES
CTRO. CULT. CDAD. DE BS. AS.
INSTITUTO DE ING. BIOMEDICA
ESC. Nº 5 JOSE DE SAN MARTIN
CIR. SUBOF. DE LA POLICIA FED.
COL. SGDO. CORAZON DE JESUS
INST. TECNICO DE BS. AS.
ESCUELA ARG. MODELO
CIDI
INST. ESTEBAN ECHEVERRIA
INST. JOSE MANUEL ESTRADA
COLEGIO S. JUAN EVANGELISTA
ENET Nº 30
COLEGIO GRAL. BENITO NAZAR
ESC. DR. ANGEL GALLARDO
ASOCIACION ISRAELITA ARGENTINA
CESCOM
ENET Nº 25 FRAY LUIS BELTRAN
ENET Nº 11 "MANUEL BELGRANO"
INST. LA INMACULADA
UTN FAC. REGIONAL RECTORADO
ESC. Nº 24
CTRO. DE INF. PSICOPEDAGOGICA
INST. SUDAMERICANO MODELO
LEICESTER COMPUTACION
ESC. Nº 15 "GRAL. J.G. ARTIGAS"
INST. CAPACITAR DE E. BALLEJOS
NTRA. SRA. DE LA MISERICORDIA
ESC. REP. ORIENTAL DEL URUGUAY
ESC. Nº 9 JOSE INGENIEROS
LICEO CULTURAL DE INFORMATICA
COL. NTRA. SRA. DE LOURDES
CAPACITAR
ESC. Nº 10 D.E. 13
ESC. MODELO D. F. SARMIENTO
INST. NTRA. SRA. DE LOS REMEDIOS
COLEGIO NTRA. SRA. DEL MILAGRO
INST. PRIV. SAN CAYETANO
LEICESTER COMPUTACION
COLEGIO SAN GREGORIO
COL. MARIE MANDOGIAN
ESCUELA Nº 11
COLEGIO SANTA BRIGIDA
ESC. Nº 14 FRANCISCO BEIRO
INST. SAN JOSE
INST. SAN VICENTE DE PAUL
ESC. Nº 11 POR LA NIÑEZ
COMERCIAL Nº 19 JUAN MONTALVO
INSTITUTO BAYARD
LAB. DE COMP. CLINICA Y EDUC.
ESC. Nº 5 URSULA DE LAPUENTE
NEW MODEL SCHOOL
COLEGIO VICTORIA
COLEGIO SANTA TERESA DE JESUS
COLEGIO INTEGRAL MARTIN BUBER
INSTITUTO ORAL MODELO
COLEGIO ISLAS MALVINAS
COL. CHARLES DE FOUCALD
C.O.E.P.O. DTSA
COLEGIO MAXIMILIANO KOLBE
INSTITUTO SAN MATEO
PSICOLOGIA EDUCACIONAL
NTRA. SRA. DEL SAGRADO CORAZON
ESCUELA ARGENTINA 2000
COLEGIO ESTEBAN ECHEVERRIA
INST. STA. ANA Y SAN JOAQUIN
CENTRO DE COMPUTACION CLINICA
ESC. TEC. RAGGIO
BS. AS. ENGLISH HIGH SCHOOL
INST. SAN PATRICIO
COLEGIO TOMAS DEVOTO
ESCUELA Nº 19
ESC. M. N. VIOLA
ESC. Nº 14 "SAN LUIS"
E.N.E.T. Nº 13
INST. SAN PIO X
ESCUELA Nº 5 D.E. 20
INST. MARIA ANA MOGAS

BUENOS AIRES:

INST. SAN GABRIEL - ADROGUE
ESC. ENS. MEDIA Nº 4 - ALGARROBO
ESCUELA Nº 28 - AVELLANEDA
INST. FRENCH - AVELLANEDA
INSTITUTO ARGOS - AVELLANEDA
E.N.E.T. Nº 6 "DR. A. GALLARDO" - AVELL.
COLEG. M. AUXILIADORA - AVELLANEDA
E.N.E.T. Nº 5 "J. HERNANDEZ" - AVELL.
ESC. Nº 4 "D. F. SARMIENTO" - AYACUCHO
ESC. Nº 1 "D. F. SARMIENTO" - AYACUCHO
E.N.E.T. Nº 1 V. PEREDA - AZUL
ESC. Nº 10 N. SRA. DEL CARMEN - AZUL
INST. SAGRADA FAMILIA - AZUL
ESC. ENS. MEDIA Nº 4 - BAHIA BLANCA
COLEGIO DON BOSCO - BAHIA BLANCA
ESC. SUP. DE COMERCIO - B. BLANCA
INST. SUP. JUAN XXIII - BAHIA BLANCA
CTRO. DE ED. AVANZADA - B. BLANCA
SEARCH S.A. - BAHIA BLANCA
COLEG. SAN V. DE PAUL - BAHIA BLANCA
ESC. Nº 6 "JOSE B. CEREIJO" - BALCARCE
ESC. REP. ARG. Nº 1103 - BALCARCE
ESC. Nº 1 "D. F. SARMIENTO" - BALCARCE
ESC. Nº 4 "B. RIVADAVIA" - BALCARCE
ESC. Nº 3 MARCOS SASTRE - BALCARCE
COLEGIO SAN JUAN - BANFIELD
ESCUELA Nº 23 - BECCAR
ESC. PROVINCIAL Nº 3 - BECCAR
ESCUELA ALMA FUERTE - B. VISTA
ESCUELA Nº 12 - BERAZATEGUI
JBS INFORMATICA - BERAZATEGUI
ESCUELA Nº 3 - BERISSO
COMPUSISTEM - BERISSO
STA. MARIA DE LAS LOMAS - BOULOGNE
COLEGIO GODSPELL - BOULOGNE
ESC. Nº 7 "MARCELO UGARTE" - BOULOGNE
COL. STA. M. DE LAS LOMAS - BOULOGNE
ESC. Nº 5 "CNEL. E. DEL BUSTO" - BRAGADO
ESC. NAC. DE COMERCIO - BRAGADO
ESCUELA Nº 10 - BRAGADO
CENTRO DE PERFEC. DOCENTE - BRANDSEN
INST. INM. CONCEPCION - BURZACO
INST. JUAN MANTOVANI - BURZACO
ESCUELA Nº 12 - CADRET
DIREC. GRAL. DE ESCUELAS - CAMPANA
I. SUP. DE F. DOCENTE Nº 15 - CAMPANA
ESC. DE EDUC. MEDIA Nº 1 - CAMPANA
ENET Nº 1 C. SARMIENTO - C. SARMIENTO
E. Nº 9 N. S. DEL CARMEN N. - C. CASARES
E.N.E.T. Nº 1 - CARLOS CASARES
ESC. Nº 1 "J. M. ESTRADA" - C. CASARES
ESC. Nº 3 M.M. DE GÜEMES - C. CASARES
ESC. Nº 1 J. C. DE ALMIRON - C. TEJEDOR
NETWORK INFORMATICA - CASEROS
ESC. DE EDUC. MEDIA Nº 15 - CITY BELL
ESCUELA DEL SOL - COLON
ESC. NAC. NORMAL SUPERIOR - COLON
ESC. Nº 1 D. F. SARMIENTO - C. PRINGLES
ESC. ENS. MEDIA Nº 1 - CHASCOMUS
CENTRO INF. ESC. Nº 5 - CHASCOMUS
COL. CORAZON DE MARIA - CHASCOMUS
COL. J. GALO DE LAVALLE - CHASCOMUS
CTR. DE INF. - CEPROC - CHASCOMUS
E. Nº 4 G.J.S. MARTIN "N"N - DE LA GARMA
COLEGIO CRISTO REY - DOCK SUD
ESC. ENS. MEDIA Nº 5 - DON TORCUATO
ESCUELA Nº 28 - DON TORCUATO
ESCUELA Nº 14 - ESCOBAR
COLEGIO JESUS MARIA - FCIO. VARELA
INSTITUTO SAN CARLOS - FCIO. VARELA
I. N. SRA. S. CORAZON - FCIO. VARELA
INST. LA SALLE - FLORIDA
COL. CARMEN AMICHED - G. ALVEAR
E. E. MED. "FORT. MATRERO" - GRAL. CERRI
INST. GRAL. PACHECO - GRAL. PACHECO
INST. FADER - GRAL. PACHECO
UTN FAC. REG. PACHECO - GRAL. PACHECO
AC. COM. BELGRANO - GRAL. RODRIGUEZ
ESC. MARIANO MORENO - GRAND BOURG
COL. SHOLEM ALEIJEM - HAEDO
ESC. Nº 8 "B. RIVADAVIA" - HAEDO
INST. DE LOS S. CORAZONES - HAEDO
INST. DE LOS S. CORAZONES - HAEDO
INST. DE E.M. "A. BROWN" - HUANGUELEN
E.N.E.T. Nº 2 - HURLINGHAM
SOC. E. Y CULTURAL ROCA - HURLINGHAM
O. T. DE COMPUTACION - I. MASCHWITZ
ESC. EDUC. MEDIA Nº 7 - I. CASANOVA
ESC. DE EDUC. TEC. Nº 6 - I. CASANOVA
CTRO. COMUNITARIO Nº 5 - ISLA MACIEL
ESC. CRISTIANA EVANG. - ITUZAINGO
INST. PRIV. A. LINCOLN - ITUZAINGO
E.N.E.T. Nº 1 - JOSE C. PAZ
INST. GRAL. J. S. MARTIN - JOSE C. PAZ
ESCUELA SAN PEDRO - JOSE C. PAZ
ESC. N. DE COMERCIO - JOSE C. PAZ
ESCUELA DE EDUC. MEDIA Nº 2 - JUNIN
INST. SUP. E F. DOC. Nº 20 - JUNIN
COLEGIO MARIANISTA - JUNIN
ESCUELA Nº 58 - LA ESPERANZA
INSTITUTO ATENEA - LANUS
INST. ECLESTON - LANUS
ESCUELA Nº 69 - LANUS
ESCUELA Nº 54 - LANUS
ESC. Nº 12 "ANT. ARGENTINA" - LANUS
ESC. Nº 5 "M. BORDOY" - LANUS
1er. C. PRIV. COMPUTACION - L. OESTE
INSTITUTO DE HIDROLOGIA - LA PLATA
INST. SAN VICENTE DE PAUL - LA PLATA
ESC. MUNICIPAL CHASCOMUS - LA PLATA
ESCUELA MEDIA Nº 3 - LA PLATA
ESC. DE EDUC. TEC. Nº 5 SAVIO - LA PLATA
ESC. ENSEÑANZA MEDIA Nº 1 - LA PLATA
FAC. CIENCIAS VETERINARIAS - LA PLATA
FAC. CS. NATURALES - LA PLATA
INST. INV. BIOQUIMICAS - LA PLATA
ESC. CONCILIO VATICANO II - LA PLATA
COLEGIO MA. AUXILIADORA - LA PLATA
UNIV. NAC. DE LA PLATA - LA PLATA
ESC. CADETES "GRAL. IRAMAIN" - L. PLATA
INST. ANTONIO PROBOLO - LA PLATA
INSTIT. DE CRIOTECNOLOGIA - L. PLATA
INST. DE LIMNOLOGIA - LA PLATA
ESC. EDUC. MEDIA Nº 2 - LAS CARABELAS
ESC. ENS. MEDIA Nº 4 - LIBERTAD
ESC. Nº 501 - LOBERIA
ESC. AGROPECUARIA Nº 1 - LOBERIA
SIDEINCO - CTRO. CAPACITAC. - LOBOS
ASOC. M. 3 DE NOVIEMBRE - L. DE ZAMORA
ASOC. COOP. NAC. UNIDAS - L. HORNOS
COL. MARIANO MORENO - LUIS GUILLON
INST. DON ORIONE - MAR DEL PLATA
ESCUELA MAR DEL PLATA - M. DEL PLATA
COLEGIO SAN AGUSTIN - MAR DEL PLATA
ESCUELA Nº 3 - MAR DEL PLATA
ESCUELA Nº 14 - MAR DEL PLATA
INST. S. VICENTE DE PAUL - M. DEL PLATA
INST. GRAL. PUEYRREDON - M. DEL PLATA
INST. SAN NICOLAS - MAR DEL PLATA
COLEG. "D. F. SARMIENTO" - M. DEL PLATA
CONS. LUIS GIANNEO - MAR DEL PLATA
E. Nº 5 "N. AVELLANEDA" - M. DEL PLATA
E. Nº 34 GRAL. PUEYRREDON - M. DEL PLATA
INST. PROF. UNIVERSIT. - M. DEL PLATA
ESC. MUNIC. CENTENARIO - M. DEL PLATA
INST. MARCOS SASTRE - M. DEL PLATA
COL. N. SRA. DEL CARMEN - M. DEL PLATA
INSTIT. ARG. MODELO - MAR DEL PLATA
J. DE INFANTES Nº 3 - MAR DEL PLATA
GUARDERIA Y J. DE INF. - MAR DEL PLATA
ESCUELA Nº 61 - MAR DEL PLATA
ESCUELA Nº 41 - MAR DEL PLATA
FUNDAC. B. DE COMERCIO - M. DEL PLATA
C. NAC. ENS. INFORMATICA - M. DEL PLATA
COLEGIO STELLA MARIS - MAR DEL PLATA
COLEGIO A. SCHWEITZER - M. DEL PLATA
ESCUELA Nº 67 - MAR DEL PLATA
ESCUELA Nº 62 - MAR DEL PLATA
ESCUELA Nº 27 - MAR DEL PLATA
ESCUELA Nº 31 - MAR DEL PLATA
J. INFANTES M. MANITOS - M. DEL PLATA
I. SUP. DE EST. ADMINISTRA - M. DEL PLATA
C.E.F.A. - MAR DEL PLATA
INST. S. VICENTE DE PAUL - M. DEL PLATA
JARDIN DE INFANTES Nº 2 - M. DEL PLATA
ESC. ENS. MEDIA Nº 5 - MARTINEZ
C. STA. TERESA DEL N. JESUS - MARTINEZ
ESC. EDUC. MEDIA Nº 2 - M. BURATOVICH
INST. JOSE M. ESTRADA - MAZA
INST. PRIVADO JUAN XXIII - MECHONGUE
ESCUELA MEDIA Nº 3 - MEDANOS
CONSEJO ESCOLAR - MERCEDES
INSTITUTO RIVADAVIA - MERLO
INST. SAN PATRICIO - MERLO
INST. NTRA. SRA. DE FATIMA - MERLO
E. ED. TEC. Nº 1 GRAL. SAVIO - MORENO
E.N.E.T. Nº 1 - MORENO
LEICESTER COMPUTACION - MORON
INST. DE ENSEÑANZA SUPERIOR - MORON
INST. SAINT THOMAS BECKET - MUNRO
INST. ARG. DE IDIOMAS - NECOCHEA
ESCUELA Nº 42 - NECOCHEA
ESC. ARG. DANESA A. MIRA - NECOCHEA
ESC. Nº 14 H. IRIGOYEN - NECOCHEA
E.N.E.T. Nº 1 - 9 DE JULIO
INST. TEC. ALIMENTARIA - 9 DE JULIO
ESC. Nº 3 "JUAN B. ALBERDI" - 9 DE JULIO
ESC. Nº 3 - 9 DE JULIO
ESCUELA Nº 17 - OLAVARRIA
COL. CTRO. CULTURAL ITALIANO - OLIVOS
COL. ASUNCION DE LA VIRGEN - OLIVOS
ESC. Nº 31 DR. A. ACUÑA - OLIVOS
COLEGIO TARBUT - OLIVOS
INST. SUP. Nº 39 - OLIVOS
ESC. Nº 1 "V. LOPEZ Y PLANES" - OLIVOS
INST. D.F. SARMIENTO - OTAMENDI
ESCUELA Nº 28 - PALOMAR
ESCUELA Nº 38 "B. RIVADAVIA" - PEARSON
INST. JOSE MANUEL ESTRADA - PELLEGRINI
A. SUP. DE COMERC. HELLER - PERGAMINO
I. TEC. COMP. PERGAMINO - PERGAMINO
ESCUELA Nº 49 - PERGAMINO
COLEGIO LA SALLE PIGUE - PIGUE
INST. CRISTO REY - QUILMES OESTE
COL. SANTO DOMINGO - RAMOS MEJIA
ESC. ARG. DEL OESTE - RAMOS MEJIA
INST. DE ENS. SUPERIOR - RAMOS MEJIA
TALLER DE COMP. DELTA TRON - R. MEJIA
ESC. Nº 10 "J.M. ESTRADA" - R. MEJIA
INST. COM. RANCAGUA - RANCAGUA
INSTIT. SAN AGUSTIN - R. DE ESCALADA
ESCUELA Nº 16 - R. DE ESCALADA
ESCUELA Nº 30 - SALADILLO
ESC. E. TEC. FRAY L. BELTRAN - SALLIQUELO
ESC. Nº 7 D.F. SARMIENTO - SMITH
ESC. D.F. SARMIENTO - S. A. DE PADUA
COLEGIO S. FERNANDO - SAN FERNANDO
NTRA. SRA. DE LA UNIDAD - SAN ISIDRO
C. CARDENAL SPINOLA - SAN ISIDRO
ESC. Nº 1 DR. COSME BECCAR - S. ISIDRO
ESCUELA Nº 22 - SAN ISIDRO
INST. 20 DE JUNIO - SAN ISIDRO
COL. S. GREGORIO MAGNO - SAN ISIDRO
INST. MARIANO MORENO - SAN ISIDRO
INST. INCORPORADO Nº 774 - S. ISIDRO
ESCUELA MODELO - SAN JUSTO
INST. ENS. MEDIA W. MORRIS - S. JUSTO
INST. N. SRA. DE FATIMA - SAN MANUEL
CENTRO DE ESTUDIOS - SAN MARTIN
I. LA SALLE SAN MARTIN - SAN MARTIN
E.N.E.T. Nº 3 E. MITRE - SAN MARTIN
LEICESTER COMPUTACION - SAN MARTIN
I. SUP. FORM. DOCENTE Nº 42 - S. MIGUEL
ESCUELA JUANA MANSO - SAN MIGUEL
COL. PARR. STA. M. TRUJUI - SAN MIGUEL
LEICESTER COMPUTACION - SAN MIGUEL
E. N. S. DE LA ASUNCION - SAN MIGUEL
I. SAN NICOLAS DE BARI - SAN NICOLAS
E.N.E.T. Nº 1 - TANDIL
ESC. Nº 28 CNEL. ESTOMBA - TAPALQUE
APIA PARVIS - TAPIALES
COLEGIO ECLESTON - TEMPERLEY
ESC. Nº 3 BME. MITRE - TIGRE
COLEGIO SAN RAMON - TIGRE
E. Nº 17 "C. DE MADRES" - 30 DE AGOSTO
E. N. DE COM. M. BELGRANO - T. LAUQUEN
ESC. Nº 8 GRAL. BARRISO - T. LAUQUEN
ESCUELA Nº 17 - TRENQUE LAUQUEN
ESCUELA Nº 2 - TRENQUE LAUQUEN
E.N.E.T. Nº 1 - TRENQUE LAUQUEN
ESC. Nº 5 C. VILLEGAS - T. LAUQUEN
ESCUELA Nº 35 - TRENQUE LAUQUEN
U.T.N. DEL T. LAUQUEN - T. LAUQUEN
CTRO. INF. DE COMPUT. - T. LAUQUEN
E.N.E.T. Nº 1 - TRENQUE LAUQUEN
COLEGIO NACIONAL - TRENQUE LAUQUEN
ESC. AGROPECUARIA - TRES ARROYOS
E.N.E.T. Nº 1 - TRES ARROYOS
I. JESUS ADOLESCENTE - TRES ARROYOS
E.N.E.T. Nº 1 - 25 DE MAYO
ESC. MEDIA Nº 1 - 25 DE MAYO
I. NUEVA ENSEÑANZA - VICENTE LOPEZ
INST. MIGUEL HAM - VICENTE LOPEZ
ESCUELA Nº 12 - VILLA ADELINA
E. N. DE C. M. BELGRANO - V. BALLESTER
I. S. LUIS GONZAGA - VILLA FLANDRIA
E.N.E.T. Nº 1 J. NEWBERY - V. LUZURIAGA
I. N. SRA. DE LOURDES - VILLA MADERO

CATAMARCA:

S. DE C. Y T. CATAMARCA - CATAMARCA

CHACO:

UTN FAC. REG. RESISTENCIA - RESISTENCIA

CHUBUT:

INST. CREAR LOGO - C. RIVADAVIA
COLEGIO Nº 11 - C. RIVADAVIA
MAGISTERIO Nº 11 - C. RIVADAVIA
I. ENS. SEC. "G. FONTANA" - SARMIENTO
C. PRIV. COMPUTACION EDUC. - TRELEW

CORDOBA:

I. DE FORMAC. INTEGRAL - C. MARINA
COL. WILLIAM C. MORRIS - CORDOBA
VP SISTEMAS - CORDOBA
COLEGIO GABRIEL TABORIN - CORDOBA

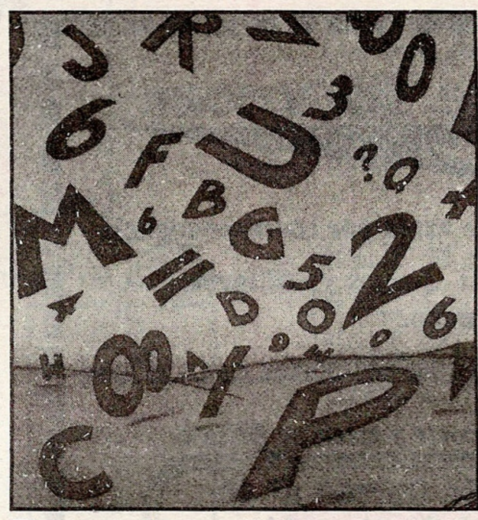
SUMARIO AÑO 2 Nº 19

COMANDANDO LA MSX



Una de las primeras "espinas" es aquella que nos clava el manual cuando empieza a hablar de las expresiones alfanuméricas. Todo muy lindo, pero... ¿qué es? **(Pág. 6)**

GEN MSX,

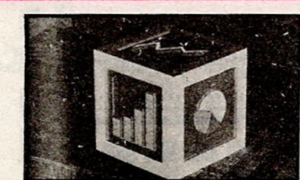


El manejo de un utilitario que permite dar los primeros pasos en la programación en código de máquina. **(Pág. 10)**

DIBUJANDO EN LA MSX

Esta nota nos introduce en el uso de instrucciones gráficas. **(Pág. 12)**

CONTABILIDAD

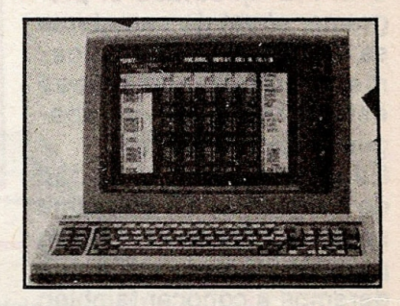


Presentamos aquí un sistema que está orientado a administrar la información de la contabilidad central. **(Pág. 16).**

RINCON DEL USUARIO

Resolvemos un problema habitual: la necesidad de imprimir texto por una impresora no MSX. Además, explicamos cómo utilizar el mouse desde el MSX BASIC. **(Pág. 17)**

BASIC DE MSX Y DE PC (2ª parte)



Ahora, aprovechando la semejanza entre ambos lenguajes, adaptamos los programas hechos en uno de ellos al otro. **(Pág. 23)**

COMO ACELERAR UN PROGRAMA (2ª parte)

Retomamos la idea de acelerar la ejecución de los programas BASIC empleando algunos trucos de programación. **(Pág. 26)**

GRAFICOS EN MSX2

Comenzamos a aprovechar las ventajas de la nueva computadora. **(Pág. 29)**

PROGRAMAS

Tipea-Mac **(Pág. 8)** - Globo **(Pág. 20)**

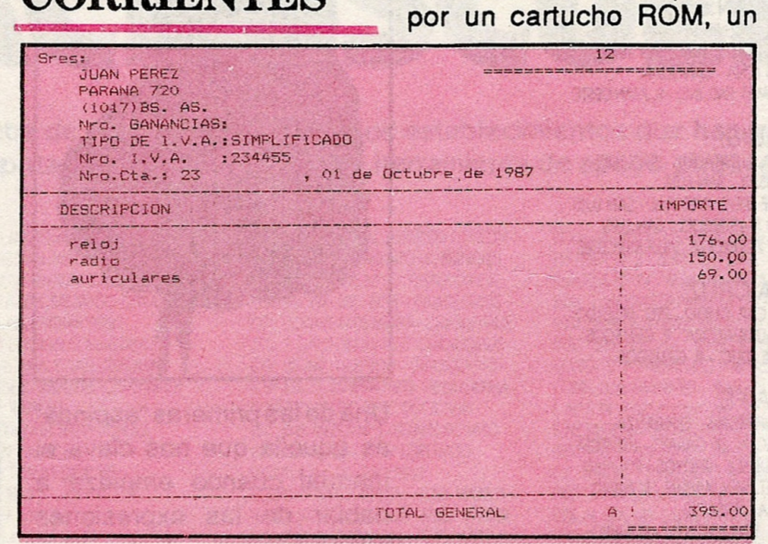
SECCIONES FIJAS

Noticias MSX **(Pág. 4)** - Soft al día **(Pág. 32)** - Buzón **(Pág. 34)**

GANADOR DEL CONCURSO MENSUAL

Eduardo Gustavo Armanino es el Segundo Ganador del Concurso Mensual Load MSX.
El trabajo por el que se hizo merecedor del premio es un programa que cambia la tipografía de las mayúsculas, las minúsculas y los números, del modo normal a la llamada letra de computadora.
Una vez ejecutado el programa, el set de caracteres se modifica automáticamente, quedando en ese estado hasta que se apague la computadora.
Eduardo vive en la ciudad de Pergamino. Su programa saldrá publicado en nuestro próximo número en el que daremos a conocer también al tercer ganador de este concurso.
Además se sabrá quién es el vencedor trimestral que recibirá un software de aplicación provisto por Telemática.

CUENTAS CORRIENTES


Sres: 12
    JUAN PEREZ
    PARANA 720
    (1017)BS. AS.
    Nro. GANANCIAS:
    TIPO DE I.V.A.:SIMPLIFICADO
    Nro. I.V.A.   :234455
    Nro.Cta.: 23   , 01 de Octubre de 1987

DESCRIPCION	IMPORTE
reloj	176.00
radio	150.00
auriculares	69.00
TOTAL GENERAL A	395.00

Computar presentó un sistema de gestión comercial para mantenimiento de cuentas corrientes de hasta 500 clientes. **Cuentas Corrientes** permite el ingreso de 3500 movimientos como máximo que reflejan la situación de la cartera de deudores. Además, realiza la impresión de comprobantes de facturas, notas de débito, de crédito y remitos. El sistema está compuesto por un cartucho ROM, un disquete "Sistema de Cuentas Corrientes", un disquete "Manual de Operación," certificado de garantía y tarjeta de registro.

Quienes deseen participar deberán dirigirse a cualquiera de las direcciones de los Centros de Asistencia para retirar los cupones y recibir las indicaciones de cómo llenarlos.

SEMINARIO

TELEMATICA inicia una serie de seminarios destinados a programadores y a casas de software donde el énfasis del trabajo no está puesto en la técnica de la programación, sino en el conocimiento y dominio de la herramienta de hardware: la computadora.
A lo largo de once sesiones de trabajo-taller, se adquirirán los conocimientos necesarios e imprescindibles para encarar de inmediato cualquier emprendimiento de desarrollo de software.
Todo programador o casa de software interesada en participar tiene la posibilidad de hacerlo con el único requisito previo de aprobar un examen de admisión.
Inscripción: TELEMATICA S.A., Chile 1347, Buenos Aires.
Los interesados del interior del país pueden consultar telefónicamente al 38-6601, de lunes a viernes de 9:00 a 13:00 y de 14:00 a 16:00 hs.



CONCURSO MENSUAL DE TELEMATICA

Se realizó un nuevo sorteo de cupones auspiciado por Telemática y los Centros de Asistencias al Usuario. **El Primer Premio** (una computadora Talent MSX DPC-200) fue para **Paz Olvera Wells**, de San Isidro. **El Segundo Premio** (un kit educativo cada uno) para **Marcela Nappe**, de Caseros, **Magdalena Funes**, de San Isidro, y **Bárbara Vazquez**, de Florida. **El Tercer Premio** (una beca para cada uno para un curso en el Centro de Asistencia a elección), **Marcelo Grigorven**, de San Isidro, **Leonel Castoriano** de Beccar, **Alejandro Revinski** de San Fernando, **Bárbara Baranowski** de Florida, y **Mariana Momano** de Olivos.
La nómina de ganadores se difundió en el programa Informática y Sociedad del Canal 3 de CableVisión.

COMANDANDO LA MSX

Una de las primeras "espinas" es aquella que nos clava el manual cuando empieza a hablar, por aquí y por allá, de las expresiones alfanuméricas. Nos enteramos que podemos manejar textos, dividir palabras y otras cosas más. Todo muy lindo pero... ¿qué es una expresión alfanumérica?



Cuando llegamos a una computadora lo habitual es tener como primer material de consulta el manual del usuario. Al comenzar a leerlo nos encontramos con términos que no comprendemos bien o no sabemos cómo aplicarlos en el universo de nuestra máquina.
Es allí donde se producen los primeros "choques" que detienen nuestro avance. Esos accidentes no sólo son normales sino que con ellos vamos aprendiendo y descubriendo trucos en el afán de superar los obstáculos.

LAS "ALFANUMERICAS"

Hablando "fácil" es una colección de caracteres. Estos caracteres pueden ser letras, números o cualquiera de los símbolos del teclado.
Estas expresiones están encerradas entre comillas ("") y ellas son, casi indefectiblemente, parte de nuestro primer contacto con la computadora. Por ejemplo, colocando PRINT "Hola, yo soy el número 1".
El manejo de estas expresiones es relativamente sencillo. Solo basta con encerrar los caracteres deseados entre comillas y anteponerles la instrucción (PRINT en este caso).

VARIABLES

Las variables alfanuméricas son lugares asignados en la memoria de la máquina que se usan para guardar datos alfanuméricos. Se distinguen por el símbolo $ al final del nombre.
La aclaración que debe hacerse es con respecto a los números. Cuando colocamos una dirección (por ejemplo "PARANA 720"), al número que forma parte de la misma no tenemos ningún interés en usarlo en una operación aritmética. Solo nos indica un dato y está relacionado con el resto del texto.

CONCATENACION

Con esta palabrita se entiende el proceso de unión de dos variables. Probemos el siguiente ejemplo:

```
10 A$="LOAD"
20 B$="MSX"
30 PRINT A$
40 PRINT B$
50 PRINT A$+B$
```

El signo "+" es utilizado para unir las dos variables que definimos al principio del programa. La concatenación es un proceso importante para darle flexibilidad y capacidad de diálogo a la computadora.

LAS FUNCIONES RELACIONADAS.

Para el tratamiento de las variables alfanuméricas existen algunas funciones. Las mismas nos permiten cortarlas y dividirlas a nuestro antojo.
Comencemos con el LEN. Esta función recuenta los caracteres que posee una variable o expresión alfanumérica y nos devuelve ese valor. La forma usual de utilizarla es colocarla dentro de una variable. Luego se usará esa variable en lo que le necesitemos. Por ejemplo:

```
10 A$="PROEDI"
20 A=LEN(A$)
30 FOR I=1 TO A
```

```
40 NEXT I
```

La línea 20 de este pequeño programa averigua la longitud de A$ y guarda ese valor en la variable A, en este caso 6.

Luego se ejecuta seis veces un lazo vacío.

LEFT$ es otra función que nos puede ser de mucha utilidad. Con la misma nos quedamos con la parte inicial de una variable alfanumérica, desechando el resto.

Agreguemos al listado anterior las líneas:

```
33 B$=LEFT$(A$,I)
35 PRINT B$
```

Ahora ejecutemos el programa y aparecerán los seis cortes de la palabra PROEDI en la primera, segunda, tercera, cuarta, quinta y sexta letra (siempre tomadas desde el principio).

Si en vez de desechar la última parte de la variable quisiéramos sacar la primera, disponemos de otra función para tal fin: RIGHT$.

Veámosla en el mismo ejemplo:

```
33 B$=RIGHT$(A$,I)
```

Y ahora el centro... Nos falta ver la función MID$. Esta función nos ofrece una porción de la variable que le damos como argumento.

Coloquemos en nuestro ya clásico ejemplo:

```
33 B$=MID$(A$,I,1)
```

En esta función debemos poner además de la variable sobre la que trabajamos, el lugar desde donde tomaremos la cadena (I), y la longitud de la misma (1, en este caso).

Con el programa como nos quedó, podemos separar en letras cualquier texto. Sólo debemos cambiar el contenido de A$ en la línea 10.

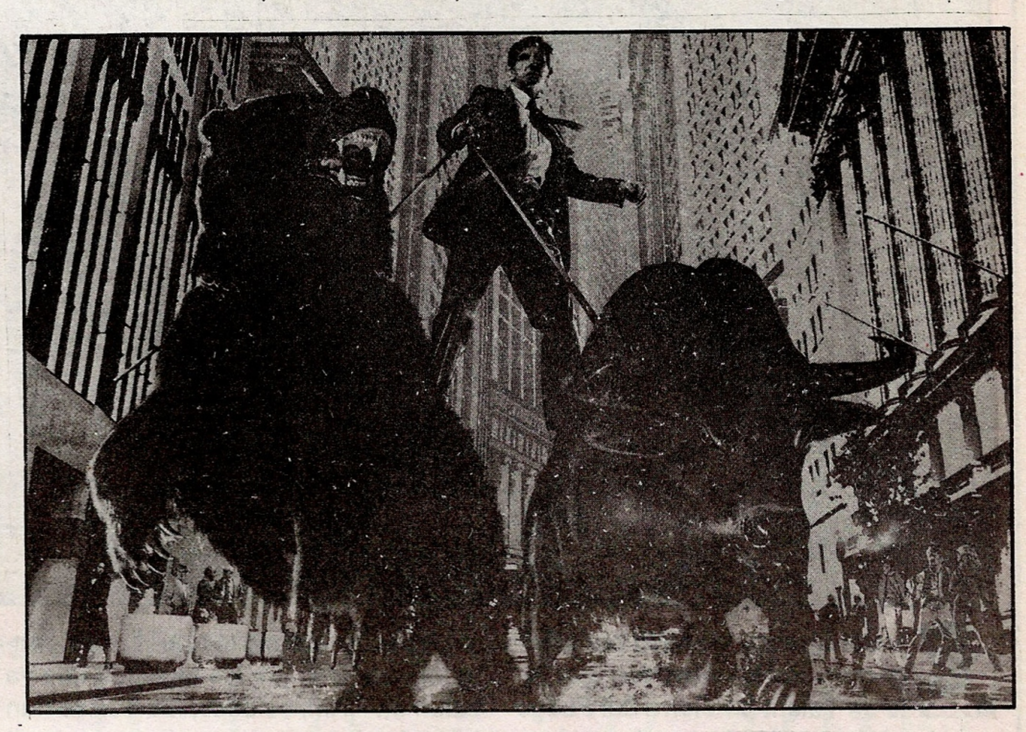
OTRAS ALTERNATIVAS

¿Qué ocurre si queremos sumar dos números que están guardados en variables alfanuméricas? Veamos:

```
10 A$="12"
20 B$="34"
30 C$=A$+B$
40 PRINT C$
```

Como vimos más arriba el signo "+" lo utilizamos para concatenar dos expresiones alfanuméricas. Por lo tanto no es de extrañar que aparezca "1234" cuando ejecutemos el programa.

Lo correcto para sumar esas dos cantidades es realizar previamente el pasaje a variables numéricas. Esto se realiza con la función VAL.



```
10 A$="12"
20 B$="34"
30 A=VAL(A$)
40 B=VAL(B$)
50 C=A+B
60 PRINT C
```

Ahora sí veremos aparecer el número 46 que es el resultado correcto de la operación pedida.

Si queremos volver a pasar una varible numérica a una alfanumérica, dicha tarea se realiza con STR$.

PARA TERMINAR...

Las variables alfanuméricas, como las numéricas, se pueden comparar unas con otras .

El "=" se cumple cuando dos variables son idénticas. Pero es interesante destacar que las relaciones ">" y "<" también pueden aplicarse, aunque con cierto cuidado.

Decimos esto porque el factor de ordenación es el código ASCII que asigna a la " A" el valor 65 y a la "Z" el 90.

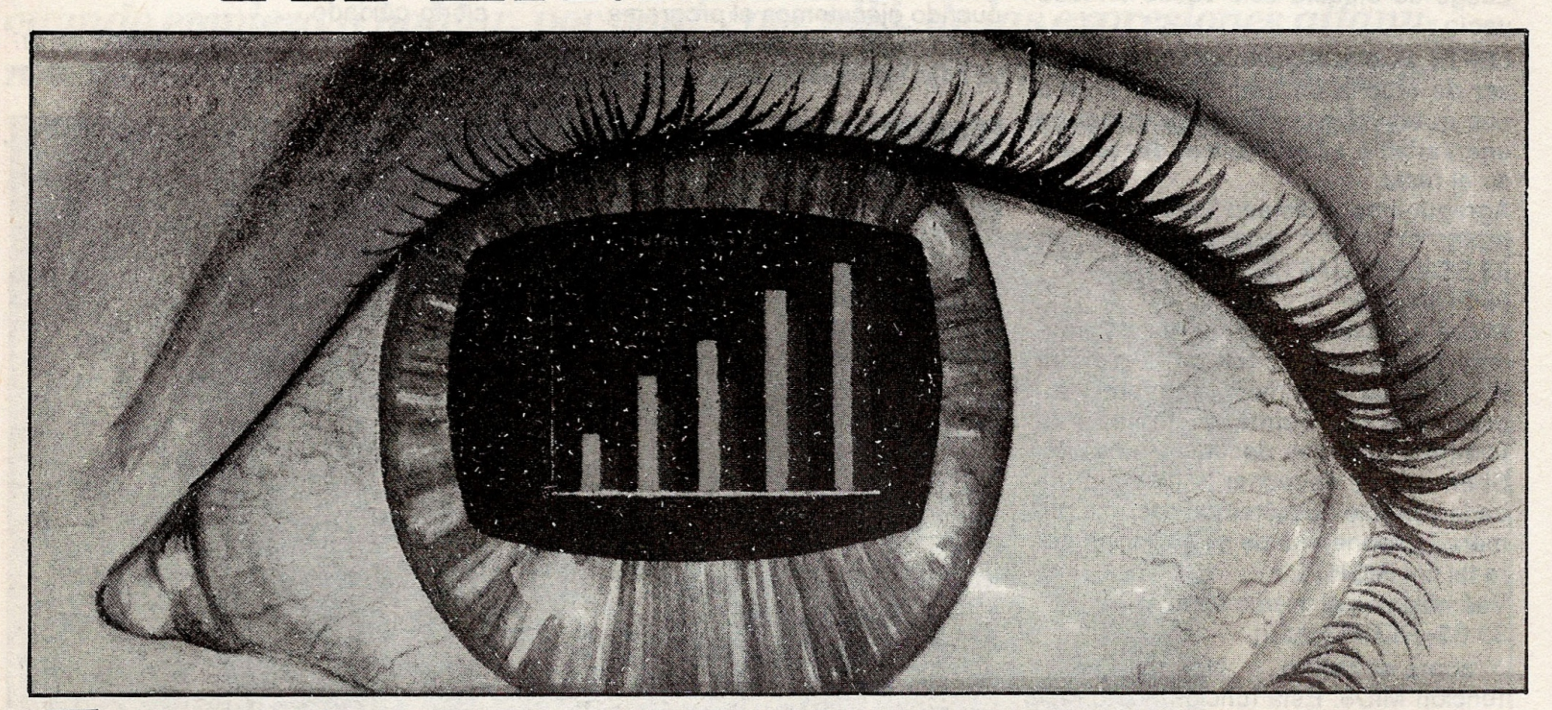
Hasta aquí no hay problemas, pero el mismo puede surgir si introducimos letras minúsculas en la variable. La "a" minúscula tiene código 97 y la "Z" el 122. Toda ordenación que queramos efectuar debe tener en cuenta si existe uniformidad en el juego de caracteres (mayúsculas o minúsculas).

Las posibilidades de "jugar" con estas funciones son enormes y esperamos que este sea un buen inicio para quienes quieran usar la computadora en ramas humanísticas y no sólo en aquellas técnicas (con números, signos y cosas por el estilo).

Fernando Pedró

TIPEA.MAC

CLASE: UTILITARIO
AUTOR: C ATASHIAN



Este programa, enviado por el Ingeniero Carlos Atashian, gerente de Talent en Tierra del Fuego y asiduo colaborador de nuestra revista, puede ser compilado con el M80 y el L80, y así generar el archivo TIPEA.COM. Este último se puede utilizar como comando interno TYPE. Realizado con el fin de que los lectores puedan profundizar sus conocimientos sobre lenguaje máquina y sistema operativo es, además, la introducción a otro programa, más complejo, que simula el comando DUMP del CP/M y que publicaremos en nuestro próximo número.

Las primeras instrucciones vuelven a ubicar el programa a partir de 4000H, lejos de donde reside el COMAND.COM, a fin de no molestarlo. Luego salta al start del programa.

Lo primero que hace es verificar (CALL VRFDRV) si el drive especificado es correcto. Para eso, lee del FCB (File Control Block) o bloque de control de archivo, el número de drive (0=drive por defecto, 1=A, 2=B, FFH=error).

Al retomar cambia la dirección de transferencia (transfer address) a 0100H, que se convierte en el comienzo de nuestra área de trabajo.

Posteriormente abre el archivo a inspeccionar y más adelante fija:

a) la extensión del registro en 1 byte.

b) el registro inicial en 0.

Seguidamente lee (o por lo menos intenta leer)1536 registros. La cantidad verdadera de registros que se han leído la retoma la subrutina en HL. De ahí en más tenemos un lazo, en el cual HL es el contador, que imprime los registros que ya están en RAM, es decir, los que se han leído del disco.

Al terminar el lazo continúa leyendo más registros hasta terminar (HL=0). Finalmente, y antes de salir, se restaura el valor original de la "Transfer address".

```
; TIPEA.MAC   Simula al comando TYPE del MSX-DOS
;
.Z80
        ASEG
        ORG     100H
;
TRAAD1  EQU     0100H
TRAADO  EQU     0080H
SETDMA  EQU     1AH
BDOS    EQU     0005H
FCB     EQU     005CH
OPENF   EQU     0FH
RNDBRD  EQU     27H
STROUT  EQU     09H
CONOUT  EQU     02H
;
        LD      HL,ORIGEN
        LD      DE,4000H
        LD      BC,FIN-ORIGEN+1
        LDIR
        JP      4000H
;
ORIGEN:
.PHASE  4000H
;
        CALL    VRFDRV
;
        LD      DE,TRAAD1
        LD      C,SETDMA
        CALL    BDOS            ;Fija la transfer address en 100H
;
        LD      DE,FCB
        LD      C,OPENF
        CALL    BDOS            ;Abre el archivo
        OR      A               ;Acc=0 es apertura O.K. - Acc=FFH es error
        LD      DE,ERROR
        JP      NZ,IMPERR
        LD      HL,0000H
        LD      (007DH),HL
        LD      (007FH),HL
        INC     HL              ;(HL)=1
        LD      (006AH),HL      ;Fija largo del registro en 1
;
MASREG: LD      DE,FCB
        LD      HL,0600H
```

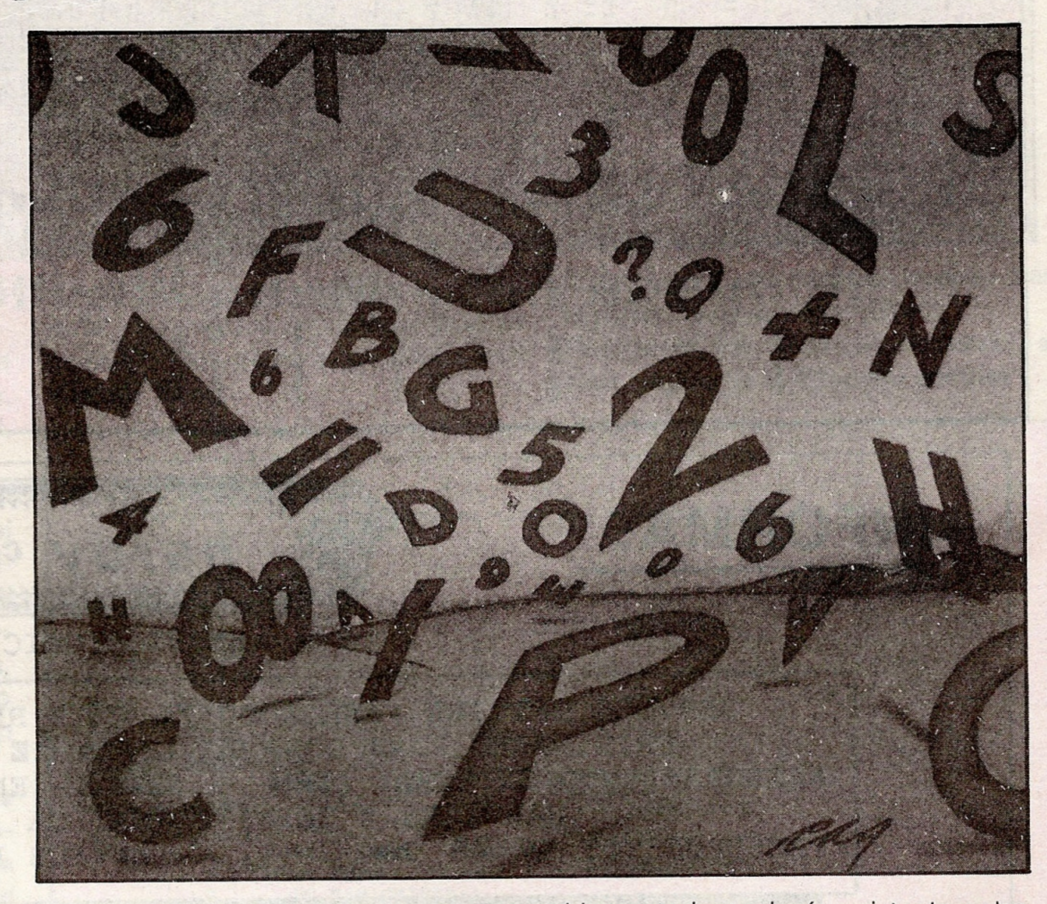
```
        LD      C,RNDBRD
        CALL    BDOS            ;Lee 0600H registros del archivo apuntado por DE
                                ;y los transfiere a la transfer address (0200H).
                                ;En HL está la verdadera cantidad de registros
                                ;que se leyeron
        LD      A,L
        OR      H               ;Se fija si terminó, es decir (HL)=0
        JR      Z,SALIDA
        LD      DE,TRAAD1
;
IMPRAM: LD      A,(DE)          ;Lee de RAM el byte a imprimir
        INC     DE
        CP      1AH             ;Se fija si es ^Z (EOF)
        JR      Z,SALIDA
        CALL    IMPRIM
        DEC     HL              ;Decrementa el contador de impresión
        LD      A,L
        OR      H
        JR      NZ,IMPRAM       ;Si el contador no es cero, toma caracter sig.
        JR      MASREG          ;Va a leer más registros
;
SALIDA: CALL    CRLF            ;Va a imprimir ret.de carro y aliment. de linea
        LD      DE,TRAADO
        LD      C,SETDMA
        CALL    BDOS            ;Restaura la transfer address original
        RET                     ;Retorna al MSX-DOS
;
;Subrutina que verifica validez del drive especificado
;
VRFDRV: LD      A,(FCB)
        INC     A
        RET     NZ
        POP     DE
        LD      DE,ERRDRV       ;Cartel de especificación inválida de drive
IMPERR: LD      C,STROUT
        CALL    BDOS
        JP      SALIDA
;
;Subrutina que imprime en pantalla el caracter leido del archivo
;
IMPRIM: PUSH    AF
        PUSH    BC
        PUSH    DE
        PUSH    HL
        LD      E,A
        LD      C,CONOUT
        CALL    BDOS
        POP     HL
        POP     DE
        POP     BC
        POP     AF
        RET
;
;Subrutina que imprime retorno de carro y alimentacion de linea
;
CRLF:   LD      DE,CRLFD
        LD      C,STROUT
        CALL    BDOS
        RET
;
;Mensajes
;
ERRDRV: DEFM 'Drive incorrecto','$'
;
ERROR:  DEFM 'Archivo inexistente','$'
;
CRLFD:  DEFB ODH,OAH,'$'
;
.DEPHASE
FIN:
;
END
```

GEN MSX - ENSAMBLADOR SIN MISTERIOS

Muchos utilitarios presentan el problema de que no pueden usarse si no se posee el manual de uso. Explicamos, en esta sencilla introducción, el manejo de un utilitario que permite dar los primeros pasos en la programación en código de máquina.

El Lenguaje Ensamblador es visto muchas veces como la solución todopoderosa para toda clase de dolencias informáticas. Estas son la rapidez de proceso, la inviolabilidad relativamente superior a cualquier otro lenguaje interpretado como es el caso del BASIC en las "home computers", además de una cierta sensación de misterio y mito que permite a sus dominadores ascender a una especie de Elite dentro del ambiente computacional.

Sin tratar de confirmar ni desmentir los conceptos anteriores, el presente artículo está destinado a permitir a aquellos que alguna vez pensaron en aventurarse por los rumbos "íntimos" del microprocesador Z80 tengan la oportunidad de comprobar por sí mismos las mencionadas características. Solamente deberán cumplir un requisito los valientes luchadores contra el byte rebelde: poseer una copia del **programa ensamblador GEN MSX**.



No son necesarios,(en absoluto necesarios) conocimientos relativos a la programación del Z80 en su propio dialecto pero si son deseables algunas ideas sólidas sobre cualquier otro lenguaje interpretado. Es frecuente encontrarse con el pensamiento erróneo de que el BASIC y el lenguaje Ensamblador pueden amalgamarse fácilmente, pero esto no es así; cada nivel de lenguaje debe utilizarse para la aplicación justa y se debe tener en cuenta que conjuntamente se arrastran sus defectos , virtudes y requerimientos.

Por ejemplo, no tiene sentido una aplicación en código de máquina que esté dedicada en su mayor parte a manejar entradas y salidas de un archivo en disco ya que el tiempo de acceso a la información depende principalmente de la rapidez del periférico y no del programa que la controla. Por el contrario, es muy aconsejable su uso en rutinas de cálculo y/o graficación. Es justamente en estos últimos campos donde se presenta otra de las dificultades de este tipo de programación que es la mayor dificultad para la puesta a punto del funcionamiento de una rutina, además de una mayor longitud equivalente, dado que no se poseen instrucciones de cálculo directas sino que deben armarse mediante traspasos, acarreos y rotaciones varias entre variables (registros) del Z80.

Precisamente el concepto de variable es uno de los que más firmemente deben tenerse en claro. Todos los registros del microprocesador se comportan como sus equivalentes de un lenguaje interpretado. Cabe recordar rápidamente la analogía existente entre una variable, numérica o alfanumérica, y un contenedor dentro del cual se deposita un número o cualquier mezcla de letras y números respectivamente en cada caso. Un registro solo puede contener un valor numérico, y no debe confundirse con la notación hexadecimal que lleva las letras de la A hasta la F incorporadas. En realidad, un registro tiene guardado su valor en forma binaria y la representación hexadecimal de su contenido es una convención con vistas a simplificar,aunque no lo parezca, su escritura. De esta manera se obvian los problemas de encolumnamiento de las cifras decimales que entre 0 y 255 varían de 1 a 3 , mientras se mantienen constantes en 2 para las hexadecimales.

TIEMBLA EL Z80

No es para menos; se acerca la hora prometida,y la vamos a iniciar con la carga del programa en cuestión. Tenemos diferentes opciones según el gusto de cada uno: para cargar el programa desde el BASIC se tecleará RUN"CAS:", con lo cual se ubicará a partir de la dirección hexadecimal 8100 (equivalente decimal 32768) ejecutándose automáticamente; o mediante BLOAD"CAS:GENMSX" que lo colocará desde la dirección hexadecimal 8800 (equivalente decimal 34816). Con esta última variante debe ejecutarse el programa manualmente con:

```
DEF  USR=&H8800
A=USR(0)
```

A este último punto de entrada se lo denomina "arranque en frío" a semejanza de un automóvil puesto en marcha por primera vez en el día. En este procedimiento se controlan e inicializan todos los parámetros y variables propios del manejo interno del ensamblador; hecho innecesario en las entradas sucesivas desde el BASIC, por lo cual se utiliza el "arranque en caliente" conseguido mediante la siguiente secuencia:

```
DEF USR=&HFFF0
A=USR(0)
```

¿PARA QUE ME HABRE METIDO?

No desesperen, las siguientes líneas nos reanimarán frente al desconcierto de no saber qué hacer para derrotar al byte rebelde que nos cambió el cursor en pantalla (¿lo había notado?).
A partir de este momento el entorno ha cambiado hacia un editor de líneas estilo CP/M (para los memoriosos expertos) y por lo tanto esto implica la imposibilidad de efectuar nuestros acostumbrados viajes y excursiones a través de la vasta planicie de la pantalla, teclas de cursor mediante.
Para comenzar a teclear una secuencia de líneas introduzca :

```
I  100,10
```

con lo cual habrá iniciado una secuencia de autonumeración automática que se incrementará, en este caso, de 10 en 10 a partir de 100.
Nuestra primera instrucción será la que indica la dirección de comienzo para depositar el código hexadecimal, es decir, traducido del mnemónico que escribiremos, al ensamblador:

```
ORG  &H9D00
```

a continuación, salvaremos en el stack del calculador los registros principales del microprocesador mediante:

```
PUSH  AF
PUSH  BC
PUSH  DE
PUSH  HL
```

Para efectuar a continuación una sencilla impresión en pantalla previa llamada a una rutina de la ROM:

```
ID  HL,0000
ID  A,65
CALL  &H004D
```

finalizando con el restablecimiento de los registros y posterior retorno a la rutina desde donde fue llamado :

```
POP HL
POP  DE
POP  BC
POP  AF
RET
```

Las líneas mencionadas deben teclearse letra por letra, finalizándolas con <ENTER>. Luego de escrito la última se abandona el modo automático presionando <CTRL> conjuntamente con <STOP>.

NO APTO PARA CARDIOPATAS

Ha llegado el momento del supremo ensamblaje. Presione la letra A seguida de <ENTER> y no empalidezca frente al cuestionamiento de "TABLE SIZE :" ya que mediante otro simple <ENTER>, el byte rebelde se conformará con preguntarnos "OPTIONS :" a lo cual responderemos muy seguros de nosotros mismos con un 4 seguido de <ENTER>. Con esto generamos el código hexadecimal a partir de la dirección especificada.
Seguramente los menos afortunados se verán en este momento ante un indescifrable mensaje de error y, lo que es peor, también en la obligación de corregirlo. Recupere el ánimo tecleando la letra L seguida de <ENTER> con lo cual verá aparecer en pantalla el listado completo del programa. Revise cuidadosamente cada línea comparando la sintaxis.
Una vez localizada la falla teclee la letra E seguida del número de línea con la falla y de <ENTER>; en la parte inferior de la pantalla aparecerá la culpable. Desplácese con las teclas de cursor a través de ella hasta colocarse encima de la falla; si su error fue un olvido oprima la tecla <INS> e inserte lo que falte. Por el contrario sobreescriba o borre lo erróneo, termine la corrección con <ENTER> y reensamble.

¡AHORA SI!

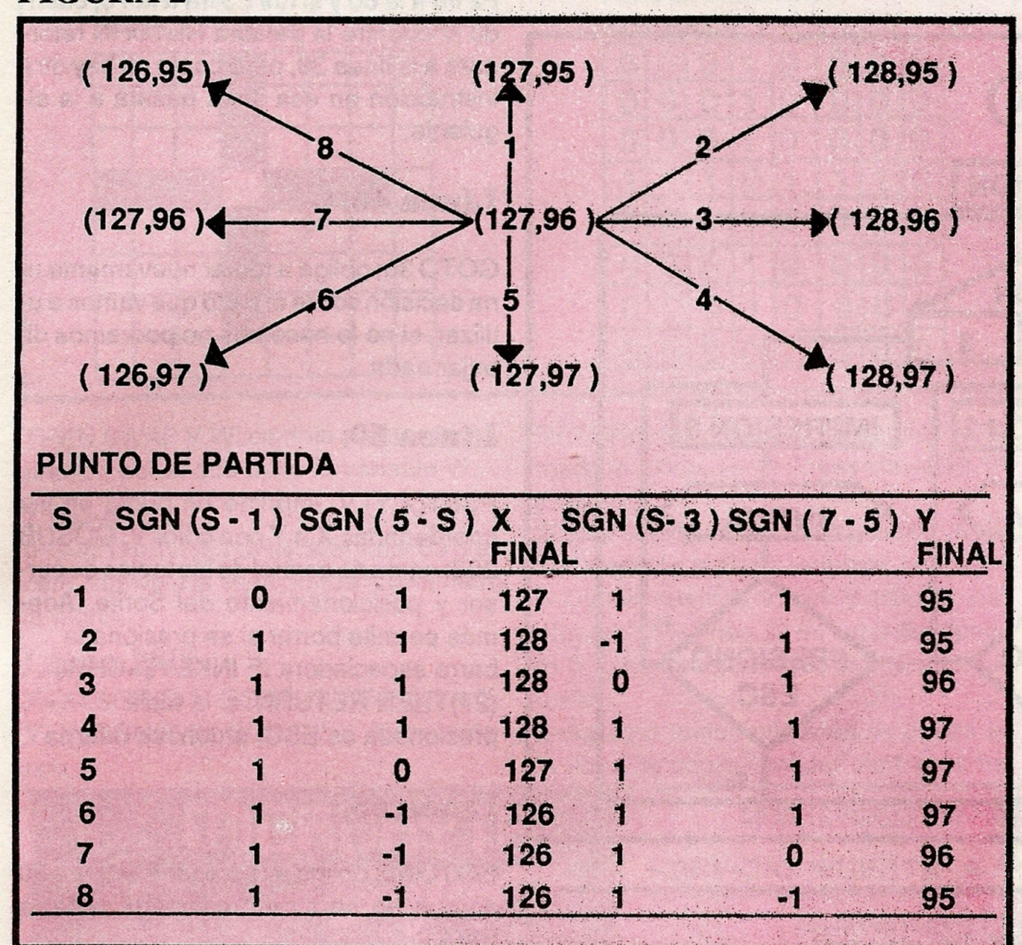
Luego de los emocionantes momentos vividos vienen tiempos más tranquilos. Una vez obtenido el código; se encuentra listo para su ejecución desde fuera del ensamblador, el cual se abandona tecleando la letra B seguida de <ENTER>. De regreso en el BASIC, nuestras ansias no podrán contenerse por observar el resultado ; teclee :

```
DEF  USR=&H9D00
A=USR(0)
```

con lo cual se verá aparecer en pantalla una maravillosa letra A. Sin tratar de apabullarlo por el resultado obtenido, la presente nota solo pretende quitarle el miedo frente al uso de lenguajes foráneos (en computación, se entiende) y animarlo en el uso de este utilitario mucho más poderoso de lo hasta aquí mostrado. ¿Lo habrá logrado?

Juan Pablo Bauer

FIGURA 2



PUNTO DE PARTIDA

S	SGN (S - 1)	SGN (5 - S)	X FINAL	SGN (S- 3)	SGN (7 - 5)	Y FINAL
1	0	1	127	1	1	95
2	1	1	128	-1	1	95
3	1	1	128	0	1	96
4	1	1	128	1	1	97
5	1	0	127	1	1	97
6	1	-1	126	1	1	97
7	1	-1	126	1	0	96
8	1	-1	126	1	-1	95

FIGURA 3

```
15 A$=CHR$(16):SPRITE$(0)=A$+A$+A$+CHR$(238)+A$+A$+A$
25 PUT SPRITE 0,(X-3,Y-4),4,0
```

una línea con origen en el primer punto y terminación en el segundo. Estos están determinados por las coordenadas entre paréntesis que figuran en la instrucción LINE. Probemos con estas tres posibilidades; no nos arrepentiremos.

Otra posibilidad es trabajar con círculos. Para ello aprenderemos dos instrucciones. La primera es CIRCLE (X,Y),5,V que nos permite dibujar un círculo de radio 5 y color V en las coordenadas X e Y. Probemos con esta instrucción en lugar de PSET Y, al correr el programa, mantengamos presionada la barra espaciadora. Para pintar el círculo debemos utilizar la instrucción PAINT(X,Y),V y colorearemos el interior del círculo, mejor dicho el área encerrada que contenga las coordenadas (X,Y), con el color V. Probemos con distintos radios y distintos colores (entre 0 y 15). IF STRIG(0)=-1 THEN SWAP V,W: si se presiona la barra espaciadora, el valor de la función STRIG(0) valdrá menos uno, por lo tanto cambiaremos el valor de V que es (blanco) 15 por el (negro) 1 y W pasará a valer 15. Si se presiona nuevamente la barra, volverá el blanco a V. SWAP intercambia los valores de las variables V y W. Para utilizar el botón del Joystick tenemos que colocar en lugar de cero: 1,2,3 o 4 según elijamos algún botón o el conector del joystick. Descubramos cuál es cuál.

Línea 30.

IF INKEY$=CHR$(27) THEN SCREEN 0:LIST: esta línea ha sido incorporada para su comodidad. INKEY$ guarda la tecla presionada, si esta coincide con CHR$(27) (la tecla ESC), entonces pasará a la pantalla de texto cero y listará el programa para poder hacer los cambios que se le sugieren. Probemos con SCREEN 1.

Línea 40.

IF S=0 THEN 20: si las teclas del cursor no han sido presionadas, el valor de S será cero, por lo tanto volverá a preguntar si se presionaron las teclas antes mencionadas en la línea 20.

Línea 50.

X=X+SGN(S-1)*SGN(5-S):SGN es la función signo que da menos uno si el número encerrado entre paréntesis es negativo, cero si es cero, y uno si es positivo. El análisis de los resultados está hecho en la figura 2. Y=Y+SGN(S-3)*SGN(7-S): Idem anterior. GOTO 20: vuelve a la línea 20.

AGREGADOS

Si aún no hemos copiado la figura 3, todavía estamos a tiempo. Estas dos líneas nos permiten ubicarnos en el dibujo a pesar de que el punto se apague, ya que utiliza figuras predefinidas llamadas Sprite. Su definición esta relacionada con los números binarios y estos con los números incluidos en la función CHR$. Para pasar un binario a decimal se puede escribir por ejemplo:

PRINT VAL ("&B00010000")
16

De esta forma se pueden transformar los números binarios de nuestro dibujo a decimales con lo que se alimentará la definición de nuestro Sprite (ver figura 6). Debemos aclarar que hay Sprites que se pueden definir con 32 números y además podemos utilizar distintos tamaños. Para interiorizarse en estos temas recomendamos releer la instrucción SCREEN y ver SPRITE$. Algunas veces, cuando manejamos, dos Sprites para dar una sensación de profundidad es necesario que un Sprite pase sobre otro, por lo tanto se maneja la idea de plano; esto último está desarrollado en la instrucción PUT SPRITE.

La línea 15.

A$=CHR$(16): se reserva en A$, el valor de CHR$(16) para evitar escribir seis veces CHR$(16). SPRITE$(0) =A$+A$+A$+CHR$(238)+A$+A$+A$: se define el Sprite cero (de 0 a 64, si son utilizados los tamaños 0 o 1, y de 0 a 255 si el tamaño es 2 o 3). Aquí se utilizan 7 CHR$ porque el último número da cero y la máquina, al no tener ese valor, lo define como CHR$(0) automáticamente.

La línea 25

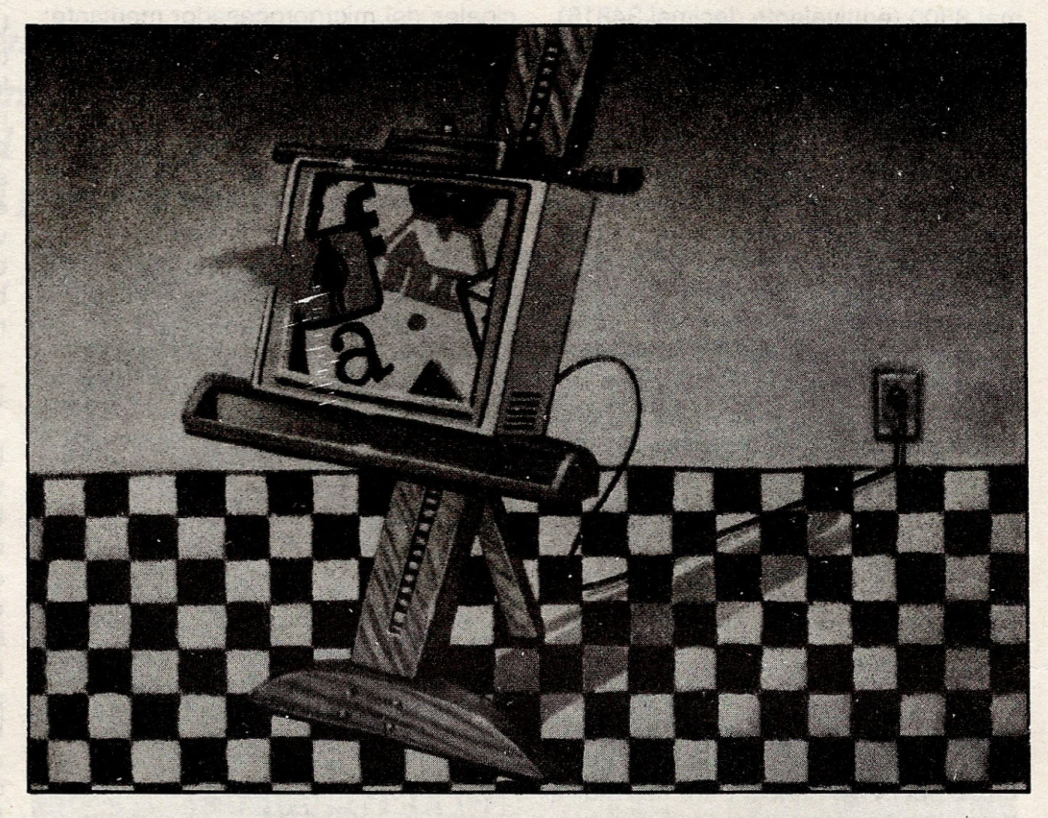
PUT SPRITE 0,(X-3,Y-4),4,0: esta instrucción nos permite ubicar el Sprite en el plano cero (de 0 a 31) en las coordenadas X-3 e Y-4 (para que el centro del Spri-

DIBUJANDO EN LA MSX

Esta nota nos introduce al uso de instrucciones gráficas. Sólo hace falta un pequeño programa,incluido en este trabajo realizado por el ganador del concurso mensual del mes de octubre.

Al ser este software muy corto existen pocas posibilidades de cometer errores, salvo los tradicionales (poner punto y coma en lugar de dos puntos, olvidarse la coma, etcétera). También se puede copiar la figura 3 que luego explicaremos.

Corregidos los errores, cuando se ejecute el programa veremos un punto en la pantalla. Este se moverá dejando un rastro si se presionan las teclas del cursor. Al presionar la barra espaciadora desaparece el punto y si hacemos pasar el punto que ha desaparecido por las líneas trazadas, ellas se borrarán. Si por el contrario no pasa por ellas, no dejará rastro. Para que vuelva a dejar rastro, hagamos aparecer el punto presionando nuevamente la barra espaciadora. También se puede dibujar con líneas punteadas manteniendo oprimida la barra espaciadora y dibujando al mismo tiempo con las teclas del cursor. Para finalizar presionemos ESC. Hasta ahora, todo fue divertido; pero puede serlo más aún cambiando algunos números o instrucciones dentro del programa. Para ello debemos conocer cómo funciona. Por lo tanto pasaremos a describirlo:



Línea 10.

COLOR: esta instrucción tiene tres argumentos, de los cuales tomaremos los últimos, color de fondo y color de borde. Cambiemos estos dos por valores comprendidos entre 0 y 15. Observaremos que si el color de fondo no es blanco (15) ni negro (1) no podremos borrar, sino que escribiremos en esos colores con el fondo que ha decidido.

SCREEN: tiene varios argumentos de los cuales vamos a ver solo el primero. Para recabar más información les recomendamos consultar el manual. El argumento mencionado nos permite elegir una entre cuatro pantallas. Las dos primeras (0 y 1) no sirven para instrucciones gráficas ya que son pantallas de texto. En cambio la 2 (alta resolución) y la 3 (multicolor) permiten el manejo de estas instrucciones. Pongamos SCREEN 3 y veremos que los puntos son más grandes porque no maneja alta resolución.

V=15: es el color de impresión. Al principio es blanco, cuando se presiona la barra espaciadora pasa a ser negro.

W=1: este número debe coincidir con el color de fondo. Si ello no sucede, en lugar de borrar imprimirá un punto visible.

X=127: es la coordenada horizontal y su valor está comprendido entre 0 y 255, que son el margen izquierdo y derecho respectivamente. Esta coordenada permite ubicar horizontalmente e inicialmente el punto. Y=96: corresponde a la coordenada vertical y sus límites son 0 y 191, que son margen superior e inferior respectivamente. Empecemos ahora en otro lugar de la pantalla.

Línea 20.

S=STICK(0): aquí se asigna el valor de la función STICK(0) que controla el conjunto de teclas de cursor y toma valores comprendidos entre 0 y 8 (ver figura 2-a)

Si se quiere, podemos usar Joystick cambiando el valor de cero encerrado entre paréntesis por un uno o un dos para el Joystick en el conector uno o dos respectivamente. Dibujemos con Joystick si tenemos uno.

PSET(X,Y),V: imprime un punto en la pantalla en las coordenadas X e Y, con el color V. Aquí tenemos varias posibilidades. Una de ellas es cambiar el punto que imprimimos con PSET por un cuadrado o rectángulo pintado, sin pintar o una línea. La instrucción LINE (X-2, Y-2)-(X+2, Y+2),V, BF nos dibujará un rectángulo de lados iguales (4 unidades de lado) con centro en las coordenadas X e Y y pintado. Si quitamos la F, no lo pintará, y si quitamos también la B, nos dibujará

FIGURA 1

```
10 COLOR 15,1,1:SCREEN 2:V=15:W=1:X=127:Y=96
20 S=STICK(0):PSET(X,Y),V:IF STRIG(0)=-1 THEN SWAP V,W
30 IF INKEY$=CHR$(27) THEN SCREEN 0: LIST
40 IF S=0 THEN 20
50 X=X+SGN(S-1)*SGN(5-S):Y=Y+SGN(S-3)*SGN(7-S):GOTO 20
```

FIGURA 4

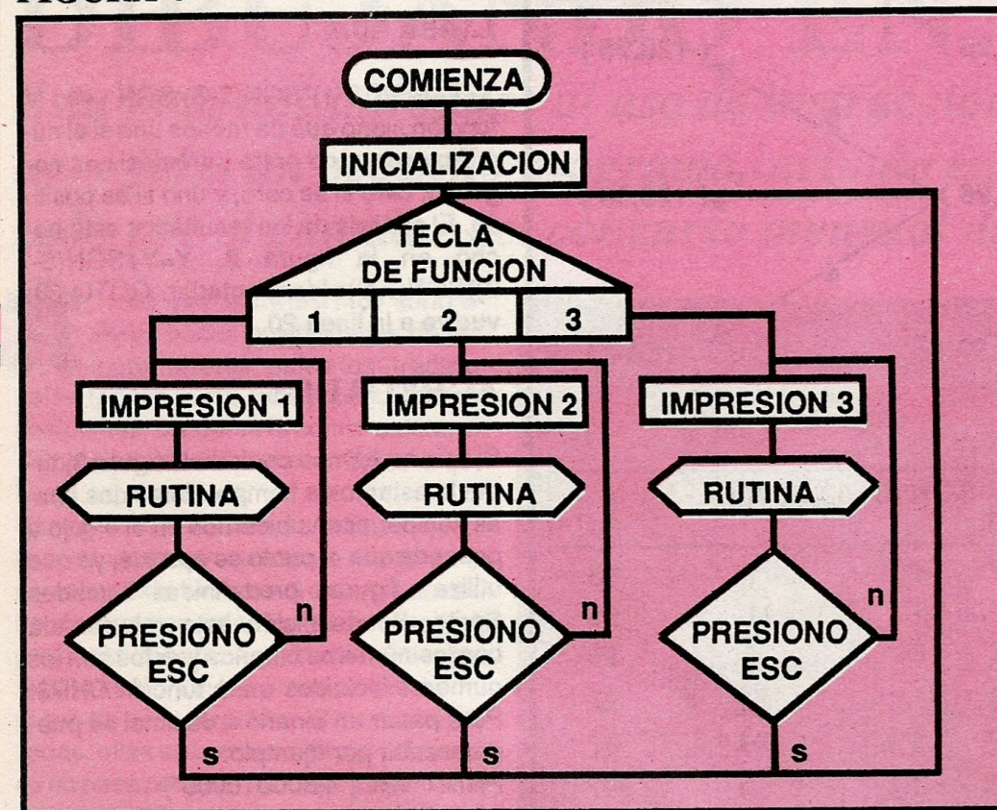


FIGURA 5

```
10 COLOR 15,1,1:SCREEN 2:V=15:W=1:X=127:Y=96
15 A$=CHR$(16):SPRITE$(0)=A$+A$+A$+CHR$(238)+A$+A$+A$
20 FOR I=1 TO 3:KEY(I) ON:NEXT
30 ON KEY GOSUB 50,60,70
40 GOTO 30
50 PSET(X,Y),V:GOSUB 200:IF INKEY$=CHR$(27) THEN RETURN
55 GOTO 50
60 LINE(X-2,Y-2)-(X+2,Y+2),V,BF:GOSUB 200:IF INKEY$=CHR$(27) THEN RETURN
65 GOTO 60
70 CIRCLE(X,Y),5,V:GOSUB 200:IF INKEY$=CHR$(27) THEN RETURN
75 GOTO 70
200 S=STICK(0):PUT SPRITE 0,(X-3,Y-4),4,0:IF STRIG(0)=-1 THEN SWAP V,W
210 IF S=0 THEN RETURN
220 X=X+SGN(S-1)*SGN(5-S):Y=Y+SGN(S-3)*SGN(7-S):RETURN
```

te coincida con el punto a imprimir), con el color 4 (azul oscuro) y el único Sprite que definimos: el cero.

TRANSFORMACIONES

Muchas personas en este momento pueden estar entusiasmadas con los descubrimientos hechos y además querrán combinar todos los efectos logrados. A ellas les dedicamos el programa en la figura 5. Este permitirá ciertas libertades para programar los efectos propios, como por ejemplo escribir con punto, círculo o cuadrado. Podremos elegir con las teclas de funciones (de 1 a 10) una forma de escribir y, cuando necesitemos otra, presionemos ESC y volvamos a elegir con las mismas teclas. También podemos escribir con puntos de distintos colores: pero siempre debemos tener cuidado de que al apretar ESC, este se encuentre precedido por el punto. (Ver figura 4).

Línea 10.

Idem anterior.

Línea 15.

Idem anterior.

Línea 20.

FOR I=1 TO 3: para cada valor de I (1,2 y 3) KEY (I) ON: prende o activa las teclas I (F1,F2 Y F3). Si queremos activar las diez teclas debemos poner en la instrucción anterior 10 en lugar de 3.
NEXT: cierra el lazo de repetición de las tres veces.

Línea 30.

ON KEY GOSUB 50,60,70: sobre el valor de la tecla de función presionada hará lo que encuentre en la línea que le indica hasta que encuentre la palabra RETURN. En otras palabras, si presionemos la tecla F1, irá a la línea 50, si fue F2 irá a la 60 y si fue F3 irá a la 70; cuando encuentre la palabra RETURN retornará a la línea 30, pero como no hay otra instrucción en esa línea pasará a la siguiente.

Línea 40.

GOTO 30: obliga a tomar nuevamente una decisión sobre el trazo que vamos a utilizar, si no lo hacemos no podremos dibujar nada.

Línea 50.

PSET(X,Y),V: imprime un punto en las coordenadas X e Y con color V. GOSUB 200: rutina de control de las teclas de cursor y posicionamiento del Sprite. Además permite borrar si se presiona la barra espaciadora IF INKEY$=CHR$(27)THEN RETURN: si la tecla presionada es ESC, entonces retorna

Línea 55.

GOTO 50: obliga a regresar a la línea 50 porque no ha sido presionada la tecla ESC.

Línea 60.

LINE(X-2,Y-2)-(X+2,Y+2),V,BF: otro efecto para poder dibujar.
GOSUB 200: ídem línea 50.
IF INKEY$=CHR$(27) THE RETURN: ídem línea 50.

Línea 65.

GOTO 60: retrocede a la 60 por las mismas razones que en la 55.

Línea 70.

CIRCLE(X,Y),5,V: otro efecto para realizar trazos.
GOSUB 200: ídem línea 50.
IF INKEY$=CHR$(27) THEN RETURN: ídem línea 50

Línea 75.

GOTO70: retrocede a la línea 70 por las mismas razones que en la línea 55.

Línea 200.

S=STICK(0): controla las teclas de cursor. PUT SPRITE 0,(X-3,Y-4),4,0: coloca el Sprite en el plano 0 y las coordenadas X-3 e Y-4 con el color 4 (azul) y el único Sprite definido el cero. IF STRIG(0)=-1

FIGURA 6

DIBUJO A REALIZAR



CODIFICACION BINARIA								CODIFICANDO PARA DEFINIR
0	0	0	1	0	0	0	0	CHR $ (16)
0	0	0	1	0	0	0	0	CHR $ (16)
0	0	0	1	0	0	0	0	CHR $ (16)
1	1	1	0	1	1	1	0	CHR $ (238)
0	0	0	1	0	0	0	0	CHR $ (16)
0	0	0	1	0	0	0	0	CHR $ (16)
0	0	0	1	0	0	0	0	CHR $ (16)
0	0	0	0	0	0	0	0	CHR $ (0)

THEN SWAP V,W: cambia el color de impresión por el que tiene la variable W.

Línea 210.

IF S=0 THEN RETURN: si el valor de S es cero, vuelve a la línea donde fue llamada la rutina.

Línea 220.

X=X+SGN(S-1) SGN(5-S): Ver figura 2.
Y=Y+SGN(S-3) SGN(7-S): ídem anterior.
RETURN: retorna a la línea donde fue llamada la rutina, estas líneas son la 50, 60 y 70.

Si queremos aumentar la cantidad de trazos distintos a elegir tenemos que:

A-Definir cuántos trazos y colocar ese número en la línea 20 en lugar del número 3. (siempre tiene que ser menos que once).

B-Agregar tantos números de líneas en la línea 30 como cantidad de trazos hallamos contado (se recomienda definirlos de diez en diez).

C-Crear las nuevas líneas con el tipo de trazo definido, luego dos puntos GOSUB 200 y, por último, luego de los dos puntos el tradicional IF INKEY$=CHR$(27) THEN RETURN

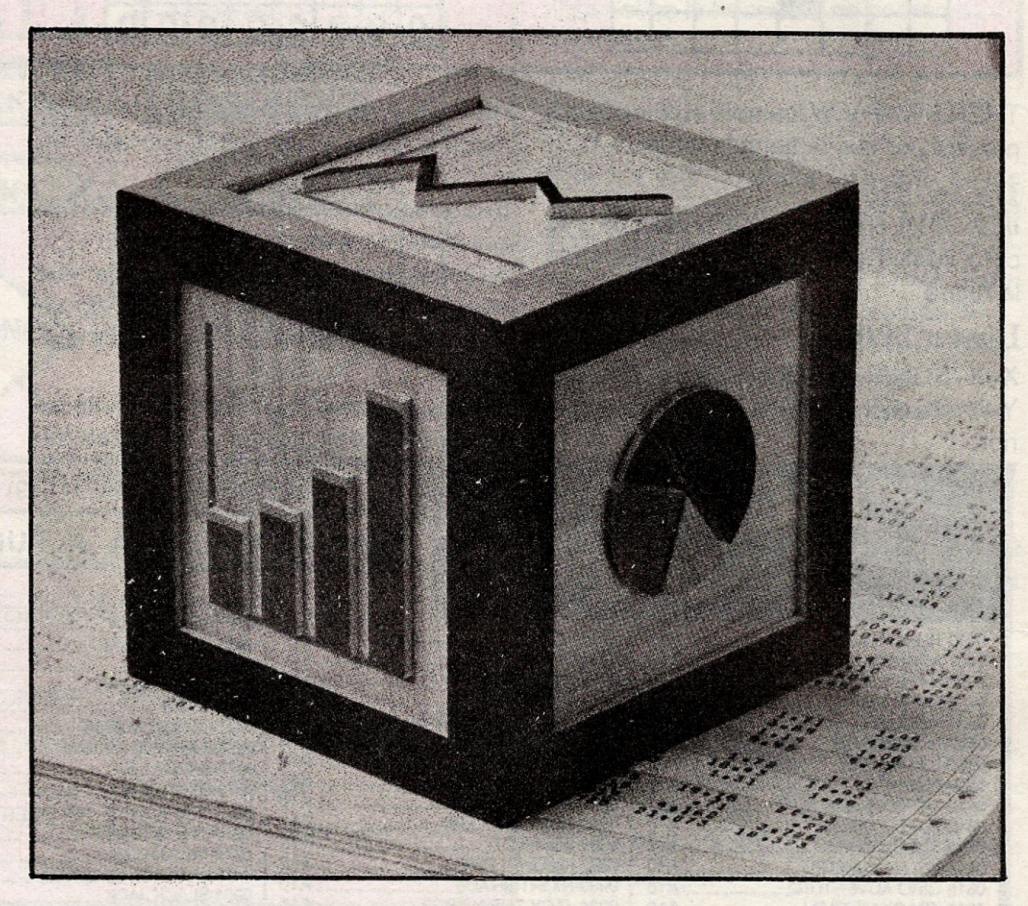
D-Colocar una línea que obligue a retroceder a las líneas creadas, una por cada línea.

Miguel Luis Kopec

CONTABILIDAD GENERAL

Presentamos aquí un sistema que está orientado a administrar la información de la contabilidad central; está hecho especialmente para los profesionales del área.

Este flexible utilitario nos presenta un Plan de Cuentas, cuya estructura permite adaptarlo a cualquier aplicación sin necesidad de reprogramación.
Al mismo tiempo, le permite al usuario ser él mismo quien codifique los asientos libremente.
Soporta cuatro niveles de clasificación, lo cual posibilita interrelaciones y la emisión de listados en forma estructurada.
Es interesante hacer notar que los movimientos que afectan los saldos de las cuentas no requieren ser ingresados en un orden determinado; en todo momento se mantienen actualizados el Saldo y la Composición del mismo para cada cuenta.
La emisión de salidas, impresas o por pantalla, puede ser solicitada en cualquier momento, sin necesidad de seguir secuencias de pasos de actualización.
Además, en todo momento el sistema permite la incorporación, o baja, de cuentas o asientos contables.
El Sistema de Contabilidad General está compuesto por un disquete con el sistema administrativo, y otro disquete utilizado para almacenar los archivos de datos del ejercicio.
Se necesita una unidad de discos y una impresora de 80 o 132 columnas.



MODO DEMOSTRACION

Hasta que no se instale, el sistema funcionará en modo demostración. Este modo nos permitirá ejercitarnos en el manejo del sistema, haciendo algunos asientos en las cuentas que creemos.
La instalación del sistema nos permitirá incorporar los datos de la empresa, del ejercicio y de la clase de impresora que utilizaremos.

SELECCION DE PROCESOS

El Sistema de Contabilidad General permite la selección de distintos procesos, mediante un menú general. El mismo tiene las siguientes opciones:

1. Plan de cuentas de mayor.
2. Procesos de asientos de mayor.
3. Emisión de listados.
4. Creación/Recreación de archivos.

La opción 1 nos permite altas de nuevas cuentas, bajas de cuentas ya existentes y modificación de los datos en una cuenta.
También podemos consultar los datos de una cuenta o hacer la emisión del plan de cuentas.
Proceso de asientos de Mayor (opción 2) permite hacer altas, bajas y consultas de datos de un asiento.
La emisión de listados se halla en la opción 3. Tenemos a nuestra disposición los listados: diario general, mayor analítico, balance de sumas y saldos, balance de presentación y estado de resultados ajustados por inflación.
Por último, la opción 4 se utiliza para la creación de nuevos archivos de datos y el pase del plan de cuentas a un nuevo ejercicio.

USO PROFESIONAL

El profesional de las ciencias económicas encontrará en este software una herramienta para sacarle el "jugo" a su computadora de norma MSX. El mismo le facilitará el trabajo contable y le abrirá el camino a nuevos clientes.
Para las pequeñas empresas también es interesante poseer este utilitario con capacidad de hasta 400 cuentas de mayor, 99 asientos contables por mes y 64 movimientos por asiento.
SDL(R) SOFTWARE presenta este utilitario en un sobre con ambos disquetes (el administrativo y el de datos).
Todo el programa está autodocumentado, por lo cual su uso es muy fácil.
Pero como si esto fuera poco, el manual del usuario está incluido en el mismo disco para ser leído por pantalla, o bien para ser impreso.
Distribuido por **Organización Maro**



COMO UTILIZAR IMPRESORAS NO MSX.

Un problema habitual entre los usuarios de la norma es la necesidad de imprimir texto por una impresora no MSX.
El problema principal que surge es que las eñes y los acentos no aparecen en el teclado de la impresora.
Es por eso que hemos generado estos programitas que permiten solucionar el problema mediante el empleo del HOOK de impresora. Veamos de qué se trata esto de los HOOKs.

Empleo de HOOKs:

El área ubicada en FD9Ah a FFC9h es la llamada "área de hooks" y se utiliza para expandir las funciones del MSX BASIC. Un hook reserva 5 bytes, que generalmente contienen 'RET'.
Cuando el MSX BASIC realiza una determinada función (por ejemplo enviar un carácter a la impresora), lo primero que hace el sistema es llamar al Hook correspondiente. Si este no está "seteado", simplemente le devuelve el control al BASIC. Sin embargo, se puede modificar el hook de tal manera que sea llamada la rutina provista por el usuario, y de esta forma, se llega a obtener el resultado deseado.
En nuestro caso, lo que debemos hacer es atrapar a las letras acentuadas y a la ñ para reemplazarlas por la secuencia <acento><retroceso><letra> ó <~><retroceso ><n>.

LISTADO 1: Programa para MSX BASIC

```
10 CLS
20 PRINT "Instalador del programa
   ACENTO"|
30 PRINT "Un momento por favor.:."
40 GOSUB 10000
50 CLS:PRINT"ACENTO instalado"
60 END
10000 RESTORE  10020:FOR  I%=&H9000
   TO  &H9084:  READ  A$:  POKE
I%,VAL("&H"+A$):NEXT
10010 DEFUSR=&H9000:I%=USR(0):RETURN
10020 DATA F3,21,B6,FF,11,75,F9,36
10030 DATA CD,23,73,23,72,21,1A,90
10040 DATA 11,75,F9,01,6C,00,ED,B0
10050 DATA FB,C9,F5,3E,7E,32,E0,F9
10060 DATA F1,FE,A4,20,06,CD,C8,F9
10070 DATA 3E,6E,C9,FE,A5,20,06,CD
10080 DATA C8,F9,3E,4E,C9,F5,3E,27
10090 DATA 32,E0,F9,F1,FE,A0,20,06
10100 DATA CD,C8,F9,3E,61,C9,FE,82
10110 DATA 20,06,CD,C8,F9,3E,65,C9
10120 DATA FE,A1,20,06,CD,C8,F9,3E
10130 DATA 69,C9,FE,A2,20,06,CD,C8
10140 DATA F9,3E,6F,C9,FE,A3,C0,CD
10150 DATA C8,F9,3E,75,C9,3A,E0,F9
10160 DATA CD,D4,F9,3E,08,CD,D4,F9
10170 DATA C9,DD,21,A5,00,FD,2A,C0
10180 DATA FC.CD,1C,00,C9
```

LISTADO 2: ACENTO para MSX-DOS

```
; RINCON DEL USUARIO TALENT MSX
; Programa acentos y e$es
; Hace que los acentos y las e$es
; puedan ser escritos por la impresora.
;
.Z80
BDOS     EQU      0005H
H.LPTO   EQU      0FFB6H  ; HOOK LPTOUT
LPTOUT   EQU        00A5H    ;  LPTOUT  (envia  un  caracter  a  la
impresora
EXPTBL   EQU      0FCC1H  ; Direccion del slot del ROM ppal.
CALSLT   EQU      001CH   ;Interslot call
MQUEUE   EQU      0F975H  ; Area reservada para PLAY (no usarlo...)
;
; Inicializacion: envia mensaje via BDOS y carga el HOOK.
        LD       DE,MENS1
        LD       C,09H
        CALL     BDOS
        DI
        LD       HL,H.LPTO
        LD       DE,START
        LD       (HL),0CDH
        INC      HL
        LD       (HL),E
        INC      HL
        LD       (HL),D
        LD       HL,BEGIN
        LD       DE,MQUEUE
        LD       BC,FIN-START
        LDIR
        EI
        LD       DE,MENS2
        LD       C,09H
        CALL     BDOS
        RET
MENS1:
        DB       'Programa cargador acento',0DH,0AH
        DB       'Un momento por favor...',0DH,0AH,'$'
MENS2:
        DB       'Programa ACENTO inicializado.',0DH,0AH,'$

BEGIN:

.PHASE  MQUEUE
;
;
;
START:
        PUSH     AF
        LD       Ã,'~'
        LD       (QUE),A
        POP      AF
        CP    164            ; Verifica si es "$"
        JR       NZ,L2
        CALL     BACK
        LD       A,'n'       ; Retorna con "$" de la impresora
        RET
L2:     CP       165         ; Verifica si es "%"
        JR       NZ,13
        CALL     BACK
        LD       A,'N'     ; Retorna con "%" de la impresora
        RET
L3:
        PUSH     AF
        LD       A,39
        LD       (QUE),A
        POP      AF
        CP       160         ; Verifica si es " "
        JR       NZ,14
        CALL     BACK
        LD       A,'a        ; Retorna con "a"
        RET
L4:     CP       130         ; Verifica si es ""
```

```
        JR      NZ,15
        CALL    BACK
        LD      A,'e'         ; Retorna con "e"
        RET
L5:     CP      161           ; Verifica si es "/"
        JR      NZ,16
        CALL    BACK
        LD      A,'i'         ; Retorna con "i"
        RET
L6:     CP      162           ; Verifica si es """
        JR      NZ,17
        CALL    BACK
        LD      A,'o'         ; Retorna con "o"
        RET
L7:     CP      163           ; Verifica si es "#"
        RET     NZ
        CALL    BACK
        LD      A,'u'         ; retorna con "u"
        RET
BACK:
        LD      A,(QUE)
        CALL    LPRINT
        LD      A,8
        CALL    LPRINT
        RET
;
; Llamada desde MSX-DOS al ROM BIOS
; LPTOUT
;
LPRINT:
        LD      IX,LPTOUT         ;Direccion del ROM BIOS
        LD      IY,(EXPTBL-1)     ;Carga direccion slot del ROM
BIOS
        CALL    CALSLT
        RET
QUE:    DB      1
FIN:
        END
```

IMPLEMENTACION:

Hemos hecho dos versiones del programa acento, uno para sistema operativo (del cual mostramos el fuente) y otro para MSX BASIC.

El programa fuente está preparado para ensamblar y "linkeditar" con el paquete M80 /L80.

En cuanto al programa BASIC, una vez ejecutado ya está disponible el uso de acentos y eñes.

Para usar el programa con MSX-WRITE, se puede hacer lo siguiente: 1) Correr el programa en BASIC y luego ingresar _WRITE.

2) Desde sistema operativo, ingresar ACENTO y luego ir al BASIC para usar el MSX WRITE como de costumbre.

NOTA 1: Si la impresora posee acentos y eñes cuyos códigos ASCII no coinciden con el de MSX, se puede cambiar el programa eliminando el llamado a la rutina BACK, y reemplazando la "n" por el código de la "ñ" en su impresora, etcétera.

NOTA 2: No utilizar este programa con el WORDSTAR ya que el mismo se encarga de imprimir los acentos y eñes.

NOTA 3: El IdeaText no permite utilizar este programa ya que el cartucho toma el control y no permite correr el programa cargador.

NOTA 4: Si usa la sentencia PLAY en BASIC, puede ocurrir un desastre... ya que la rutina reside en el área de almacenamiento del PLAY.

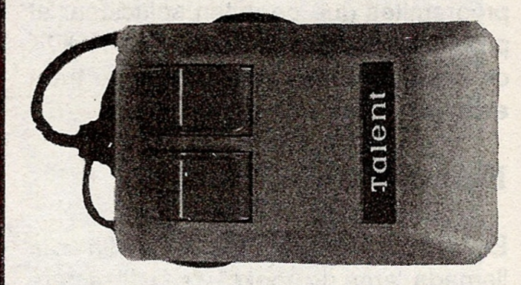
PROGRAMA 1

```
10 CLS:SCREEN1:COLOR15,1,1:GOSUB 80:GOSUB100:PUT SPRITE0,(100,100),15:X=100:Y=100
20 A%=USR(0)
30 DY%=PEEK(&HD000):DX%=PEEK(&HD001)
40 IF DX%>127 THEN DX%=DX%-256
50 IF DY%>127 THEN DY%=DY%-256
60 X=X-DX%:Y=Y-DY%
70 PUT SPRITE0,(X,Y):GOTO 20
80 FORI=1TO8:READ A$:SP$=SP$+CHR$(VAL("&H"+A$)):NEXT:SPRITE$(0)=SP$:RETURN
90 DATA 00,78,70,78,5C,4E,04,00
100 RESTORE 120:FOR I%=&HC000 TO &HC082: READ A$: POKE I%,VAL("&H"+A$):NEXT
110 DEFUSR=&HC000:RETURN
120 DATA E5,3E,0F,D3,A0,DB,A2,E6
130 DATA 8F,F6,30,5F,3E,0F,D3,A0
140 DATA 7B,D3,A1,06,10,10,FE,3E
150 DATA 0E,D3,A0,DB,A2,57,7B,E6
160 DATA 8F,5F,06,02,10,FE,3E,0F
170 DATA D3,A0,7B,D3,A1,3E,0F,A2
180 DATA 07,07,07,07,4F,06,02,10
190 DATA FE,3E,0E,D3,A0,DB,A2,57
200 DATA 7B,F6,30,5F,3E,0F,D3,A0
210 DATA 7B,D3,A1,3E,0F,A2,B1,67
220 DATA 06,02,10,FE,3E,0E,D3,A0
230 DATA DB,A2,57,7B,E6,8F,5F,3E
240 DATA 0F,D3,A0,7B,D3,A1,3E,0F
250 DATA A2,07,07,07,07,4F,06,02
260 DATA 10,FE,3E,0E,D3,A0,DB,A2
270 DATA 57,3E,0F,A2,B1,6F,22,00
280 DATA D0,E1,C9
```

USO DEL MOUSE DESDE BASIC.

Entre las consultas llegadas al Servicio de Consulta Telefónica INFO-TALENT se encuentra la duda sobre cómo poder utilizar el mouse desde MSX BASIC.

Para tal fin hemos implementado una pequeña rutina en código de máquina que obtiene los resultados de los movimientos del mouse y los deposita en las direcciones &HD000 y &HD001 para y y x respectivamente.



En el programa 1, hemos aplicado la rutina sobre un sprite en SCREEN 1 para que se note cómo se debe programar el uso de esta rutina.

Este programita permitiría (con esfuerzo considerable) generar un entorno de trabajo del tipo GEM (o sea como la Macintosh de Apple), ya que el mouse se presta para ello.

Como siempre, ante cualquier duda (que no sea "El programa me da error") pueden llamarnos al Info-Talent.

ACTIVIDADES Y SERVICIOS

El IDES -Instituto de Enseñanza Superior- anuncia una serie de actividades y servicios, con las siguientes características:

Actividades:

-Dictado de seminarios de formación docente con certificados autenticados por el Ministerio de Educación y puntaje reconocido:

• Cursos para docentes de nivel primario y secundario.

-Cursos especiales para docentes de escuelas comerciales.

- Asesoramiento a escuelas.

-Dictado de cursos y talleres de computación para niños (LOGO), adolescentes y adultos:

Capacitación para personal administrativo de empresas y profesionales.

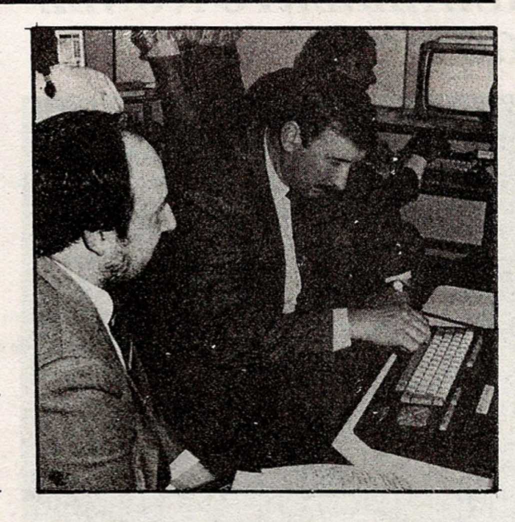
-Manejo de utilitarios:

• Base de datos, procesador de palabras, planilla electrónica, etcétera.

-Servicios gratuitos a la comunidad:

• Otorgamiento de becas para cursos de introducción a la informática que se dictan en forma regular en las instalaciones del Instituto.

Para mayor información los interesados pueden dirigirse a: Bolívar 55 Ramos Mejía, Tel. 658-4777, de lunes a viernes en los horarios de 9 a 13 y de 14 a 21 hs y los sábados de 9 a 13 horas. Otras Sedes: Belgrano 160, Morón, Mendoza 2728, B 1º Belgrano, Cap. Fed.



CENTROS DE ASISTENCIA AL USUARIO DE TALENT MSX

CAPITAL FEDERAL

Centro Cultural de la Ciudad de Buenos Aires
Taller Logo de computación
Junín 1930
Martes a Sábados de 15 a 19.30 horas

Fundación de Informática y Educación
Centro de Computación Clínica
Asistencia al Usuario Discapacitado
Ramsay 2250 - Pabellón F
Tel. 784-2018
Lunes a Viernes de 8 a 17 horas

Barrio Norte
Uriburu 1063 - Tel. 83-6892/826-6692
Lunes a Viernes de 9 a 21 horas
Sábados de 9 a 12 horas

Belgrano
Mendoza 2728 - Tel. 781-2271
Lunes a Viernes de 15 a 22 horas

Centro
Av. Córdoba 654 - Tel. 392-5328/7611/8043/8051/8251
Lunes a Viernes de 12 a 21 horas
Sábados de 9 a 13 horas

Flores
Gral. Artigas 354 - Tel. 612-3902
Lunes a Viernes de 14 a 20 horas
Sábados de 10 a 13 horas

Palermo
Guatemala 4733 - Tel. 71-4124
Lunes a Viernes de 14 a 21 horas
Sábados de 9 a 13 horas

San Telmo



Chile 1345 - Tel. 37-0051 al 54
Lunes a Viernes de 10 a 13 y de 14 a 19 horas

GRAN BUENOS AIRES

Lanús
Caaguazú 2186 - Tel. 247-0678
Lunes a Viernes de 9 a 13 y de 16 a 20 horas
Sábados de 9 a 13 horas

Morón
Belgrano 160 - Tel. 629-3347
Lunes a Viernes de 9 a 13 y de 14 a 21 horas
Sábados de 9 a 13 horas

Ramos Mejía
Bolívar 55 - 1er. piso - Tel. 658-4777
Lunes a Viernes de 9 a 13 y de 14 a 21 horas
Sábados de 9 a 13 horas

San Isidro
Av. Centenario 705 - Tel. 743-9678/747-6094
Lunes a Viernes de 9 a 21 horas
Sábados de 9 a 12 horas

Vicente López
Av. Maipú 625 - Tel. 797-6720
Lunes a Viernes de 10 a 19 horas

INTERIOR DEL PAIS

La Plata - Pcia.de Bs. As.
Calle 48 No. 529 - Tel. (021) 249905 al 07
Lunes a Viernes de 9 a 21 horas
Sábados de 9 a 13 horas

Bahía Blanca - Pcia.de Buenos Aires
Gral. Paz 257 - Tel. (091) 31582
Lunes a Viernes de 9 a 12 y de 16 a 20 horas

Córdoba - Pcia.de Córdoba
9 de julio 533
Lunes a Viernes de 8 a 12 y de 16 a 20 horas

Villa María - Pcia.de Córdoba
Corrientes 1159 - 2do. piso - Tel (0535) 24311
Lunes a Viernes de 16 a 23 horas
Sábados de 8 a 12 y de 15 a 18

Mendoza - Pcia.de Mendoza
Rivadavia 76 - 1er. piso - Tel. (061) 291348/293151
Lunes a Viernes de 8 a 13 y de 16 a 20 horas
Sábados de 8 a 13 horas

Santa Fe - Pcia.de Santa Fe
Rivadavia 2553 Loc.22 - (042) 41832
Lunes a Viernes de 9 a 12 y de 16 a 19 horas
Sábados de 9 a 12 horas

Rosario - Pcia.de Santa Fe
Barón de Mau 1088
Lunes a Viernes de 8 a 12 y de 15 a 19 horas
Sábados de 9 a 12 horas

S.M.de Tucumán - Pcia.de Tucumán
Bolivar 374 - Tel. (081) 245007
Lunes a Viernes de 9 a 18 horas

EL GLOBO

AUTOR: ISMAEL ALBERTO NUÑEZ
CLASE : JUEGO



La idea del juego es simple; tres sobrevivientes de un naufragio han llegado a una isla y debemos ir a rescatarlos con nuestro globo. Tendremos que hacer tres viajes pues no podemos traer a más de uno por vez.

Los inconvenientes que deberemos sortear son de dos tipos. Primero hay que evitar que los pájaros que cruzan la pantalla choquen con nuestro globo, pues este está inflado con un gas muy volátil y en el momento de la colisión explota. Segundo debemos controlar el combustible, que se muestra en la esquina inferior izquierda de la pantalla, ya que si se nos acaba, finaliza el juego.

Podemos recargar el mismo volando sobre el depósito de combustible, marcado con la letra F, y arrojando bolsas sobre ese lugar, tarea que se realiza con la barra espaciadora. El método de navegación es aprovechar las corrientes de aire, que tienen distintos sentidos y velocidades según la altitud. Para elevar el globo podemos utilizar un joystick o, en su defecto, la tecla de cursor arriba; la caída se produce por gravedad.

Luego de cumplir nuestra primera misión, con el rescate de los tres náufragos obtenemos una condecoración y pasamos al nivel siguiente, en donde la dificultad aumenta ya que hay más pájaros y éstos vuelan más bajo.

El juego termina cuando obtenemos cinco condecoraciones. Entonces nos podremos sentir héroes de la aeronavegación.

ESTRUCTURA DEL PROGRAMA

10-80: comienzo
80-440: creación de sprites
450-790: rutina principal
800-1070: subrutina de la caída de las bolsas
1080-1200: subrutina de colisión de sprites
1210-1290: subrutina de impresión de pantalla
1300-1350: subrutina de elección de la fuerza del viento
1360-1550: preparación de pantalla
1560-1660: final
1670-1690: recogida del náufrago
1700-1760: náufrago salvado
1770-1830: muerte del náufrago por explosión
1840-1910: tercer náufrago salvado
1920-1960: subrutina de borrado de sprites
1970-2030: músicas
2040-2300: presentación
2310-2420: instrucciones
2430-2440 : datos de sprites

VARIABLES

A: rescate de un náufrago
M: nro. de sprite del náufrago tomado
Q,W: coordenada de los pájaros
PX: rescate de los tres náufragos
X,Y: coordenadas del globo
D: stick
CX: variable de disparo
F: combustible
V(I): fuerza del viento
J,G,I,T: auxiliares de lazo
FE: aumenta combustible
G: vidas
I$: inkey$
R,C: coordenadas auxiliares
H: contador de explosiones
RC: número de condecoraciones
CO: impresión de condecoración
SS: impresión de náufrago salvado
B: número de náufragos salvados
MU$,ME$,
MF$,MR$,: música
MT$

```
10 REM###############EL GLOBO############
20 REM###############1985###############
30 REM###############I. Nuñez############
40 REM##################################
50 CLEAR2000
60 STOPON:ONSTOPGOSUB2470
70 GOTO 1970
80 GOSUB1210
90 REM######SPRITE######
100 FORT=1TO8
110 READB$
120 H$=H$+CHR$(VAL(B$))
130 NEXTT
140 SPRITE$(2)=H$
150 FORT=1TO8
160 READF$
170 K$=K$+CHR$(VAL(F$))
180 NEXTT
190 SPRITE$(13)=K$
200 FORT=1TO32
210 READA$
220 S$=S$+CHR$(VAL(A$))
230 NEXTT
240 SPRITE$(1)=S$
250 FORT=1TO32
260 READD$
270 P$=P$+CHR$(VAL(D$))
280 NEXTT
```

[redacted]

```
290 FORI=4TO8:SPRITE$(I)=P$:NEXT
300 FORT=1TO32
310 READC$
320 L$=L$+CHR$(VAL(C$))
330 NEXTT
340 SPRITE$(3)=L$
350 FORT=1TO32
360 READE$
370 M$=M$+CHR$(VAL(E$))
380 NEXTT
390 FORI=10TO12:SPRITE$(I)=M$:NEXT
400 FORT=1TO32
410 READR$
420 U$=U$+CHR$(VAL(R$))
430 NEXTT
440 FORI=14TO18:SPRITE$(I)=U$:NEXT
450 REM ####LAZO PRINCIPAL####
460 IFA=0THEN480
470 PUTSPRITEM,(X+4,Y+20),1,13
480 PUTSPRITE4,(Q,W),13,4
490 PUTSPRITE5,(Q-30,W+20),6,5
500 PUTSPRITE6,(Q-15,W+40),13,6
510 PUTSPRITE7,(Q+4,W+70),4,7
520 IFPX=1THENPUTSPRITE8,(Q-20,W+95),4,8
530 PUTSPRITE1,(X,Y),1,1
540 D=STICK(0)ORSTICK(1)
550 IFSTRIG(0)ORSTRIG(1)THENCX=1:GOSUB810
560 Q=Q-2:IFQ<1THENQ=255
570 IFD>0ANDD<3ANDY<3THENY=3
580 IFD>1ANDD<5ANDX>254THENX=254
590 IFY>143THENY=144
600 IFD=1THENX=X:Y=Y-3:F=F+.15:LINE(10,18
0)-(F,190),15,BF:PLAY"V602L64G"
610 IFF>50THEN1560
620 Y=Y+1
630 PUTSPRITE4,(Q,W),13,14
640 PUTSPRITE5,(Q-30,W+20),6,15
650 PUTSPRITE6,(Q-15,W+40),13,16
660 PUTSPRITE7,(Q+4,W+70),4,17
670 IFPX=1THENPUTSPRITE8,(Q-20,W+95),4,18
680 PSET(160,160),15
690 IFY>0ANDY<20THENX=X+V(1)
700 IFY>30ANDY<50THENX=X-V(2)
710 IFY>70ANDY<90THENX=X+V(3)
720 IFY>110ANDY<130THENX=X-V(4)
730 IFX>240THENX=240
740 IFX<1THENX=1
750 IFX>145ANDX<160ANDY>144ANDA=1THEN460
760 IFX>145ANDX<160ANDY>144THENGOSUB1670
770 IFA=1ANDX>13 ANDX<29 ANDY>144THENGOSU
B1700
780 PSET(160,160),1
790 GOTO 460
800 REM #### CAIDA DE BOLSA ####
810 FORJ=Y+16TO130STEP6
820 IFA=1THENGOTO1060
830 Q=Q-1:Y=Y-.5:IFY<3THENY=3
840 IFX>152ANDX<164ANDJ>115ANDJ<125THEN10
30
850 PUTSPRITE2,(X+4,J),6,2
860 PUTSPRITE1,(X,Y),1,1
870 PUTSPRITE4,(Q,W),13,4
880 PUTSPRITE5,(Q-30,W+20),6,5
890 PUTSPRITE6,(Q-15,W+40),13,6
900 PUTSPRITE7,(Q+4,W+70),4,7
910 IFPX=1THENPUTSPRITE8,(Q-20,W+95),4,8
920 FORI=1TO35:NEXT
930 PUTSPRITE4,(Q,W),13,14
940 PUTSPRITE5,(Q-30,W+20),6,15
950 PUTSPRITE6,(Q-15,W+40),13,16
960 PUTSPRITE7,(Q+4,W+70),4,17
970 IFPX=1THENPUTSPRITE8,(Q-20,W+95),4,18
980 SOUND7,254:SOUND8,15
990 SOUND0,J:SOUND1,0
1000 NEXTJ
1010 SOUND8,0
1020 GOTO 1050
1030 FE=F-3:IFFE<10THENFE=10
1040 LINE(FE,180)-(50,190),5,BF:F=FE
1050 PUTSPRITE2,(250,130),0,2
1060 SOUND8,0:CX=0:SX=0
1070 RETURN
1080 REM ###CHOQUE SPRITES###
1090 IFX>165ANDY>127THEN1200
1110 SOUND0,0:SOUND1,0:SOUND6,40:SOUND7,1
4:SOUND8,10:SOUND9,20:SOUND11,130:SOUND12
,150:SOUND13,1
1120 SPRITEOFF:G=G-1
1130 PUTSPRITE1,(X,Y),1,3
1140 FORT=1TO1000:NEXT
1150 X=22:Y=145
1160 PUTSPRITE1,(X,Y),1,1
1170 IFA=1THENPLAY"AGFDEADFFDAFD":GOSUB17
70
1180 A=0
1190 IFG=0THENLINE(218,180)-(256,190),15,
BF:GOTO 1560ELSE1490
1200 RETURN
1210 REM #### GRAFICO ####
1220 SCREEN2,2:COLOR1,15,1:CLS
1230 CIRCLE(120,0),20,10,3.14,0:PAINT(121
,1),10:CIRCLE(40,10),20,5,,,.3:PAINT(40,1
0),5
1240 CIRCLE(30,20),15,5,,,.3:PAINT(30,20)
,5:CIRCLE(115,50),10,5,,,.3:PAINT(115,50)
,5
1250 CIRCLE(110,56),10,5,,,.4:PAINT(110,5
6),5:CIRCLE(175,49),20,5,,,.4:PAINT(175,4
9),5
1260 CIRCLE(185,31),20,5,,,.3:PAINT(185,3
1),5:CIRCLE(235,61),10,5,,,.3:PAINT(235,6
1),5
1265 LINE(0,110)-(255,175),5,BF:DRAW"C8BM
0,140R70G35L40U35":PAINT(1,141),8
1270 DRAW"C12BM120,175R100E30L100G30":PAI
NT(152,174),12:CIRCLE(160,160),16,15,,,.3
5:PAINT(160,160),15:CIRCLE(165,120),7,15:
PAINT(165,120),15
1280 PSET(163,117),15:PRINT#1,"F":LINE(0,
176)-(256,191),15,BF:CIRCLE(160,160),8,2,
,,.35:PAINT(160,160),2: DRAW"C15BM14,164R
20E10L20G10D5R20NU5E10U5":PAINT(16,163),1
5
1290 CO=119:RC=0:PLAYME$,MI$
1300 REM ###FUERZA DEL VIENTO###
1310 E=RND(-TIME)
1320 FORI=1TO4
1330 V(I)=INT(RND(1)*4+1)
1340 IFV(I)<=1ORV(I)>4THEN1330
1350 NEXT
1360 REM ###PREPARA PANTALLA###
1370 M=9:S=0:SS=0:B=0:A=0
1380 X=22:Y=145:Q=250:W=10:F=10:IFPX=1THE
NG=3ELSEG=4
1390 LINE(0,175)-(256,175),1
1400 LINE(86,175)-(86,191),1
1410 LINE(177,175)-(177,191),1
1420 LINE(125,175)-(125,191),1
1430 LINE(F,180)-(50,190),5,BF
1440 PSET(56,182):PRINT#1,"FUEL"
1450 R=175:C=157:H=0
1460 PUTSPRITEM+1.(R,C),1 .10
```

```
1470 PUTSPRITEM+2,(R+15,C),1 ,11
1480 PUTSPRITEM+3,(R+30,C),1 ,12
1490 PSET(180,182),15:PRINT#1,"VIDAS";
1500 LINE(218,180)-(256,190),15,BF
1510 FORI=1TOG:PSET(213+I*8,182),15:PRINT
#1,"A":NEXT
1520 SPRITEON:ONSPRITEGOSUB1080
1530 STOPON:ONSTOPGOSUB2470
1540 ON ERROR GOTO 1560
1550 RETURN
1560 REM #####FIN########
1570 PLAYMF$:FORI=1TO900:NEXT
1580 GOSUB1920
1590 LINE(0,176)-(256,191),10,BF
1600 PSET(32,182):PRINT#1,"DESEAS JUGAR C
TRA VEZ (S/N)?";:I$=INPUT$(1)
1610 I$=INKEY$:IFI$="S"ORI$="s"THEN1640
1620 IFI$="N"ORI$="n"THENBEEP:END
1630 GOTO 1610
1640 PX=0:LINE(0,176)-(256,191),15,BF
1650 PUTSPRITE8,(200,209),0,18:GOSUB1290
1660 GOTO 460
1670 REM ####RECOGIDA#####
1680 PLAYMR$:A=1:M=M+1
1690 RETURN
1700 REM ######SALVADO######
1710 PLAYMR$:S=S+10
1720 PUTSPRITEM,(80+S,179),7 ,13:A=0
1730 B=B+1:IFB=3THEN1840
1740 IFH=2ANDB=1THENGOSUB1450
1750 IFH=1ANDB=2THENGOSUB1450
1760 RETURN
1770 REM ####SIN DESTINO####
1780 H=H+1:SS=SS+5
1790 PUTSPRITEM,(10+SS,200),0,13
1800 IFH=2ANDB=1THEN1450
1810 IFH=1ANDB=2THEN1450
1820 IFH>2THENGOSUB1450
1830 RETURN
1840 REM #####3 SALVADOS#####
1850 PLAYMT$:CO=CO+10:PSET(CO,180),15
1860 PRINT#1,"L":FORI=1TO1000:NEXT
1870 PX=1:RC=RC+1
1880 GOSUB1920
1890 IFRC=5THEN1910ELSEGOSUB1300
1900 GOTO 460
1910 FORI=1TO1000:NEXT:LINE(0,176)-(256,1
91),4,BF:PSET(65,182),15:PRINT#1,"MASIE P
AL COCO TIO":FORI=1TO3000:NEXT:GOTO 1590
1920 REM ####ANULA SPRITES####
1930 FORI=1TOM
1940 PUTSPRITE9+I,(200,208),0,10
1950 NEXT
1960 RETURN
1970 REM#####MUSICA######
1980 MU$="V9T10002D"
1990 ME$="O4FEDC":MI$="O4GGBO5CO4"
2000 MF$="O4L10FE-D-E-FFF"
2010 MR$="T50S1M500008CD"
2020 MT$="T150E3D3C3D3E3E3E3"
2030 PLAY"DCO3A#O4CDDD"
2050 REM#####PRESENTACION#####
2060 COLOR 15,1,1:SCREEN2,,0,2:OPEN"GRP:"
AS#1
2070 DRAW"C10BM6,40E22R10G5L5G4R4G3L2G5R1
0G5L17":PAINT(14,39),10
2080 DRAW"C11BM27,40E22R6G17R10G6L15":PAI
NT(41,39),11
2090 DRAW"C8BM5,84E64R25G10L12G40R17E10L6
E6R15G30L38":PAINT(44,83),8
2100 DRAW"C2BM50,84E64R15G50R30G14L45":PA
INT(65,83),2
2110 DRAW"C7BM130,84E64R30G30R12G34L42BM1
56,71E18R16G18L16BM181,44E14R9G14L9":PAIN
T(145,83),7
2120 DRAW"C3BM179,84E64R40G64L40BM214,71L
10E34R3":LINE(240,38)-(215,71),3:PAINT(20
0,83),3
2130 CIRCLE(143,38),39,13
2140 PAINT(143,38),13
2150 LINE(116,55)-(128,130),13:LINE(170,5
5)-(148,130),13
2160 LINE(130,66)-(135,125),13:LINE(155,7
0)-(145,125),13
2170 LINE(128,130)-(148,145),15,BF
2180 DRAW"C15BM128,130E5R20G5BM153,125D15
G5"
2190 PRESET(25,160):PRINT#1,"POR"
2200 DRAW"BM70,175R3E3U28E2R2F3L2F3L2F3L2
F3L2F3L2F3L2F3L2F3L2F3L2R2E2U28E3R3BM
100,155D15F3R6E3U15BD14D1F3R1BR5U17BD4E3R
6F3D13BU21G2H3L2G3L2"
2210 DRAW"BD19BR20R8BL8H3U13E3R4F3D5G3L4H
2BU8BR14E3R8D2L1D2L1D2L1D2L1D2L1D2L1D2L1D
2L1D2L1D2L1D1R8E3BD2BR10E3R2F3R2E3BL7U28B
R5BU4G3L2H3L2G3BR20BD33R2U2L2D2E2"
2220 PSET(115,28),13:PRINT#1,"instruc-":P
SET(122,38),13:PRINT#1,"ciones."
2230 PSET(124,50),13:PRINT#1,"(s/n)?";
2240 I$=INPUT$(1)
2250 IFI$="S"ORI$="s"THEN2280
2260 IFI$="N"ORI$="n"THEN80
2270 PLAYMU$:GOTO 2240
2280 SCREEN0:CLS:COLOR 15,4,5:WIDTH(40)
2300 PRINTTAB(12)"INSTRUCCIONES":PRINT
2310 PRINT"█ Hay que posar el GLOBO en el
 centro      del circulo y regresar al pun
to de       partida con el naufrago.":PRIN
T
2320 PRINT"█ Salvando tres naufragos pasa
ras a la     fase siguiente,y conseguiras
una con-     decoracion [L].":PRINT
2330 PRINT"█ Para que suba el GLOBO pulsa
 [↑]":PRINT
2340 PRINT"█ El GLOBO es arrastrado por c
orrientes     de aire.":PRINT
2350 PRINT"█ Para reponer FUEL hay que de
jar caer      una bolsa en el deposito [F],
pulsando      [SPACE].":PRINT
2360 PRINT"█ Hay que evitar el contacto c
on los        pajarracos."
2370 PRINTTAB(12)"█  INKEY  █"
2380 I$=INKEY$:IFI$=""THEN2380
2390 GOTO 80
2400 DATA16,16,56,56,68,124,124,124
2410 DATA152,152,252,26,25,36,36,36
2420 DATA3,7,15,10,13,10,15,7,10,9,4,4,2,
3,3,3,192,224,240,176,80,176,240,224,80,1
44,32,32,64,192,192,192
2430 DATA3,59,59,29,14,7,3,3,2,57,95,251,
1,0,0,0,128,192,224,240,240,112,112,112,2
41,239,30,252,254,0,0,0
2440 DATA165,105,69,168,69,36,210,45,65,1
8,8,4,4,7,7,7,36,74,20,99,20,106,129,108,
4,144,32,64,64,192,192,192
2450 DATA3,5,7,2,1,15,22,38,68,3,3,2,2,2,
6,0,128,64,192,128,0,224,208,200,68,128,1
28,128,128,128,192,0
2460 DATA0,0,3,7,13,1,1,59,95,251,1,0,0,1
,3,7,0,0,128,192,192,192,193,63,252,222,2
24,224,224,224,192,128
2470 BEEP:RUN
```

EL BASIC DE MSX Y EL DE PC (2º parte)

Continuamos la serie iniciada en el número pasado. Ahora, aprovechando el parecido entre ambos lenguajes adaptamos los programas hechos en uno de ellos al otro.



Por el parecido entre ambos lenguajes podemos adaptar los programas hechos en uno de ellos al otro.
Veremos las instrucciones DRAW, SOUND y PLAY, que tienen un tratamiento que difiere un poco de sus equivalentes MSX ya que en las computadoras personales no se necesitan con tanta frecuencia gráficos o sonidos. Se las define como macros, es decir, que tienen sus propios comandos para poder usarlas.
La MSX tiene catorce comandos diferentes en MSX BASIC que le permiten dibujar en ocho direcciones variadas, con un punto de referencia absoluto (el 0,0 de la pantalla) o relativo (la coordenada del último pixel iluminado), moverse sin dibujar, dibujar y retornar al punto inicial, inicializar en qué cuadrante lo hará, definir el color y determinar la escala. Amén de poder llevarle estos parámetros a través de una variable. El modo de usarla ya fue explicado en otros artículos; lo que nos importa ahora es su compatibilidad con BASICA(el basic de IBM) o cómo podemos lograrla.

DRAW, UN LAPIZ ELECTRONICO

DRAW en IBM tiene 15 comandos, catorce son similares a los descriptos anteriormente y hay uno más que permite rellenar la figura generada. Tal es la similitud que podemos establecer la tabla 1 de equivalencias:
A simple vista parece directamente compatible, en cierto modo lo es, salvo que BASICA permite otra forma de escritura; por ejemplo: M+=x;,=y; donde x e y son las coordenadas.

SOUND

En el caso de nuestra computadora no permite poner en los registros del PSG (programable sound generator) directamente el valor de las notas, envolventes, etcétera.
Distinto tratamiento tienen las P.C. ya que carecen de sintetizador y deben generar sonidos por otros medios. El formato que tienen es:
SOUND f, d (f-FRECUENCIA, d-DURACION)
Esta instrucción permite emitir un tono de altura y duración programables; la frecuencia se da directamente en HERTZ y tiene un rango que va desde 37Hz hasta 32767Hz redondeándose al entero más próximo. Como dato ilustrativo diré que un LA equivale a 432Hz. La duración está medida en ciclos del reloj interno de la máquina, por ejemplo un segundo equivale a 18 ciclos, un minuto a 1092 y 65535 ciclos son practicamente una hora. Con valores inferiores a uno se producirán efectos curiosos.

TABLA 1: EQUIVALENCIAS

	MSXBASIC	BASICA
ARRIBA	U	U
ARRIBA DERECHA	E	E
DERECHA	R	R
DERECHA ABAJO	F	F
ABAJO	D	D
ABAJO IZQUIERDA	G	G
IZQUIERDA	L	L
IZQUIERDA ARRIBA	H	H
DIBUJA LINEAS	M	M
COLOR	C	C
MOVER SIN DIBUJAR	B	B
MUEVE Y RETORNA	N	N
ANGULO (4 cuadrantes)	A	A
ESCALA 0-255	S	S
RELLENO DE FIGURA	-	P

PLAY

La MSX tiene ocho comandos que permiten definir: las notas que se van a emitir, la octava correspondiente, la longitud pausa, tempo, volumen, modulación y envolvente de cada uno de los tres canales con que cuenta el chip de sonido. También, como en DRAW, estos parámetros se pueden transmitir como una variable.
Como hemos dicho anteriormente BASICA carece de chip de sonido pero pese a todo se las ingenia para ofrecer una instrucción de múltiples recursos ya que dispone de una macroinstrucción similar a la que conocemos. Ver la La tabla 2 de equivalencias.

COMPATIBILIDAD

BASICA dispone de tres comandos que le permiten programar los matices de ejecución: ejecución normal dado por MN, las notas terminan con un pequeño silencio de1/8 entre cada nota. Con el comando ML (ligado) se obtiene un sonido ligado; las notas picadas se obtienen con el comando MS (stacatto) que introduce silencios de 1/4 de nota entre cada una. Observemos que la lógica esta sofisticada al máximo en comparación con el hardware, que no permite matices ni en la intensidad sonora ni en el timbre. ML y MS, utilizadas con habilidad, pueden dar lugar a cierto tipo de composición armónica.

Ahora estudiemos las instrucciones semicompatibles que vimos en la primera parte.

La selección de la modalidad texto o modo 0 se realiza por medio del comando SCREEN 0. Este es el mismo que selecciona la página visible (entre 4 u 8), y la página activa donde se escribe, aunque no necesariamente, la misma. Se puede dar la impresión de una presentación instantánea, preparando la página siguiente antes de hacerla visible:

SCREEN 0, c , pe , pv

Otra modalidad es SCREEN 1, que permite dibujar sobre una cuadrícula de 200 x 320 pixels con 4 colores.

Y la modalidad final es el modo 2 de alta resolución, en el que se utiliza un bit por elemento de imagen, permitiendo así, una resolución de 640 x 200 pixels, pero con solo dos colores.

Vemos que en general es una instrucción compatible si se toman las precauciones de verificar el formato.

WIDTH también debe considerarse ya que toma tres formas de instruccionamiento :

WIDTH n — Fija la visualización de la pantalla (40 o 80 columnas)

WIDTH #c , a — Determina la " anchura " del periférico (c =canal ; a = ancho)

WIDTH "lpt1",a — Establece el ancho a imprimir

En PAINT se agregan un parámetro más a la instrucción y adquiere así la sintaxis PAINT (x , y), cll, cli; siendo este último color del límite de la figura en cuestión.

Hasta aquí vimos someramente la compatibilidad de ambos BASIC llegando a la conclusión de que podemos emplear, con las limitaciones del hardware, un programa hecho en PC en una MSX y viceversa. Pero para demostrarlo, aquí hay uno que existe en el disco DEMO de las PERSONAL COMPUTERS.

El listado original es el listado 1.

Como comentario final se puede agregar que no solo se deben verificar las instrucciones y los parámetros sino que también hemos de tener en cuenta (sobre todo con 40 columnas y 192 x 256 pixels), las limitaciones de nuestra máquina y así poder mantener la estética con la que originalmente cuenta.

TABLA 2 EQUIVALENCIAS

	MSXBASIC	BASICA
NOTAS	A- a G+ o G#	A- a G+ o G#
OCTAVAS	O(1-8)	O(0-6)
NOTAS ABSOLUTAS	N(0-96)	N(1-84)
TEMO	T(32-255)	T(32-255)
SILENCIOS	R(1-64)	P(1-64)
LONGITUD	L(1-64)	-
VOLUMEN	(0-15)	-
MODULACION	M(1-65535)	-
FORMA	S(1-15)	-

LISTADO 1

```
10      REM * BIORHYTHM PROGRAM *
20      WIDTH "lptl:",132
30      REM FROM INTERFACE AGE, AUGUST, 1976
40      REM EDITED & TESTED BY JIM PETERSEN ON 89/27/78
50      REM %% PROGRAM COMPUTES HOW CERTAIN PERSON WILL FEEL PHYSICALLY,
60      REM    EMOTIONALLY AND INTELLECTUALLY ON ANY DAY %%
70      DIM A(12),M$(12)
80      DATA 0,31,59,90,120,151,181,212,243,273,304,334
90      FOR I=1 TO 12:READ A(I):NEXT
100      PI=3.14159
110     CLS:REM Screen clear character
120     X=6:Y=0
130     PRINT:PRINT
140     LOCATE X,Y:X=X+1:PRINT "This program will print out your personal
BIO-RYTHM analysis for any number of"
150     LOCATE X,Y:X=X+1:PRINT "days you select.Dates should be entered in
numerical form with the MONTH, DAY "
160     LOCATE X,Y:X=X+1:PRINT "and YEAR separated by commas. I.E."
170     PRINT
180     INPUT "How many days do you want plotted";S
190     IF S<=0 THEN 180
200     LOCATE X,Y:X=X+1:PRINT "What is the date at which you would
the";S;"day"
210     INPUT "analysis to start";M,D,Y
220     S=S-1
230     M=INT(M):D=INT(D):Y=INT(Y)
240     INPUT "What is the date of your birth";MB,DB,YB
250     MB=INT(MB):DB=INT(DB):YB=INT(YB)
260     INPUT "What is your name";A$
270     P=-1
280     DEF FNI(X)=INT(20*SIN(2*PI*X/33)+.5)
290     DEF FNE(X)=INT(20*SIN(2*PI*X/28)+.5)
300     DEF FNP(X)=INT(20*SIN(2*PI*X/23)+.5)
310     T=INT(D+365.25*Y+A(M)+.01*M-.03)
320     TB=INT(DB+365.25*YB+A(MB)+.01*MB-.03)
330     X=T-TB
340     V=INT((40-LEN(A$))/2)
350     RB=TB-1-INT((TB-1)/7)*7
360     DATA MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY,SUNDAY
370     FOR I=0 TO 6:READ DAY$(I):NEXT
380     DATA MON,TUE,WED,THU,FRI,SAT,SUN
390     FOR I=0 TO 6:READ D$(I):NEXT
400     DATA JANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JULY,AUGUST
410     DATA SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
420     FOR I=1 TO 12:READ M$(I):NEXT
430     IF P THEN LPRINT SPC(2):FOR I=1 TO V:LPRINT "*";:NEXT
440     IF P THEN LPRINT "BIO-RHYTHM ANALYSIS FOR ";A$;
450     IF P THEN LPRINT "**************************";
460     IF P THEN LPRINT:LPRINT
470     R=T-1-INT((T-1)/7)*7
480     LE=21+LEN(DAY$(R))+LEN(M$(M))+LEN(STR$(D))+LEN(RIGHT$(STR$(Y),2))
490     V2=INT((64-LE)/2)
500     IF P THEN LPRINT SPC(2):FOR I=1 TO V2:LPRINT "*";:NEXT
510     IF P THEN LPRINT "STARTING DATE: ";DAY$(R);", ";M$(M);STR$(D);", 19";
520     PRINT RIGHT$(STR$(Y),2);:COLOR 7,0
530     IF P THEN LPRINT RIGHT$(STR$(Y),2);
540     IF P THEN LPRINT "******************";
550     IF P THEN LPRINT:LPRINT
560     LL=LEN(DAY$(RB))+LEN(M$(MB))+LEN(STR$(DB))+LEN(RIGHT$(STR$(YB),2))
570     LB=LL+18
580     VB=INT((64-LB)/2)
590     IF P THEN LPRINT SPC(2)
600     IF P THEN LPRINT "*************** BIRTH DATE: ";DAY$(RB);",
";M$(MB);STR$(DB);", 19";
610     IF P THEN LPRINT RIGHT$(STR$(YB),2);
620     IF P THEN LPRINT "*****************"
630     IF P THEN LPRINT:LPRINT:LPRINT
```

```
640   IF P THEN LPRINT TAB(20);"DOWN",TAB(39);"CRITICAL",TAB(62);"UP"
650   IF P THEN LPRINT D$(R);D;M$(M);
660   F(0)=42:F$(0)="*"
670   F(1)=42+FNI(X):F$(1)="I"
680   F(2)=42+FNE(X):F$(2)="E"
690   F(3)=42+FNP(X):F$(3)="P"
700   FOR I=0 TO 2
710   FOR J=I+1 TO 3
720   IF F(I)<F(J) THEN 770
730   IF F(I)=F(J) THEN F(I)=0:F$(I)="":F$(J)="X":GOTO 770
740   Q=F(I):Q$=F$(I)
750   F(I)=F(J):F$(I)=F$(J)
760   F(J)=Q:F$(J)=Q$
770   NEXT J:NEXT I
780   FOR I=0 TO 3
790   IF P THEN LPRINT TAB(F(I));F$(I);
800   NEXT
810   X=X+.5:IF P THEN LPRINT
820   IF INT(X)<>X THEN 660
830   IF Z=S THEN 900
840   R=R+1:IF R=7 THEN R=0
850   D=D+1:IF M=4 OR M=6 OR M=9 OR M=11 THEN L=30:GOTO 890
860   IF M=2 AND Y/4=INT(Y/4) THEN L=29:GOTO 890
870   IF M=2 THEN L=28:GOTO 890
880   L=31
890   IF D>L THEN D=D-L:M=M+1:GOTO 910
900   IF P THEN LPRINT D$(R);D;:Z=Z+1:GOTO 660
910   IF M=13 THEN M=1:Z=Z+1:GOTO 630
920   Z=Z+1:IF D=1 THEN GOTO 630 ELSE GOTO 660
930   IF P THEN LPRINT:LPRINT
940   IF P THEN LPRINT CHR$(12)
950   INPUT "Do you want another Biorhythm";A$:IF LEFT$(A$,1)="Y" OR
LEFT$(A$,1)="y" THEN CLEAR 2000:P=0:GOTO 70
960   END
```

Versión adaptada a MSX :

```
10    REM * BIORHYTHM PROGRAM *
20    WIDTH 40:KEYOFF
30    REM FROM INTERFACE AGE, AUGUST, 1976
40    REM EDITED & TESTED BY JIM PETERSEN ON 89/27/78
50    REM %% PROGRAM COMPUTES HOW CERTAIN PERSON WILL FEEL PHYSICALLY,
60    REM    EMOTIONALLY AND INTELLECTUALLY ON ANY DAY %%
70    DIM A(12),M$(12)
80    DATA 0,31,59,90,120,151,181,212,243,273,304,334
90    FOR I=1 TO 12:READ A(I):NEXT
100    PI=3.14159
110   CLS:REM Screen clear character
120   PRINT:PRINT
130   X=3:Y=0
140   LOCATE Y,X:X=X+1:PRINT  "This program will print out your perso -"
150   LOCATE Y,X:X=X+1:PRINT  "nal BIO-RYTHM analysis for any number of"
160   LOCATE Y,X:X=X+1:PRINT  "days you select.Dates should be entered "
170   LOCATE Y,X:X=X+1:PRINT  "in numerical form with the MONTH,DAY and"
180   LOCATE Y,X:X=X+1:PRINT  "YEAR separated by commas. I.E."
190   PRINT
200   INPUT "How many days do you want plotted";S
210   IF S<=0 THEN 200
220   LOCATE Y,X:X=X+1:PRINT  "What is the date at which you would"
230   LOCATE Y,X:X=X+1:PRINT  "the";S;"day"
240   INPUT "analysis to start";M,D,Y
250   S=S-1
260   M=INT(M):D=INT(D):Y=INT(Y)
270   INPUT "What is the date of your birth";MB,DB,YB
280   MB=INT(MB):DB=INT(DB):YB=INT(YB)
290   INPUT "What is your name";A$
300   P=-1
310   DEF FNI(X)=INT(20*SIN(2*PI*X/33)+.5)
320   DEF FNE(X)=INT(20*SIN(2*PI*X/28)+.5)
330   DEF FNP(X)=INT(20*SIN(2*PI*X/23)+.5)
340   T=INT(D+365.25*Y+A(M)+.01*M-.03)
350   TB=INT(DB+365.25*YB+A(MB)+.01*MB-.03)
360   X=T-TB
370   V=INT((40-LEN(A$))/2)
380   RB=TB-1-INT((TB-1)/7)*7
390   DATA MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY,SUNDAY
400   FOR I=0 TO 6:READ DAY$(I):NEXT
410   DATA MON,TUE,WED,THU,FRI,SAT,SUN
420   FOR I=0 TO 6:READ D$(I):NEXT
430   DATA JANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JULY,AUGUST
440   DATA SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
450   FOR I=1 TO 12:READ M$(I):NEXT
460   IF P THEN LPRINT SPC(2):FOR I=1 TO V:LPRINT "*";:NEXT
470   IF P THEN LPRINT "BIO-RHYTHM ANALYSIS FOR ";A$;
480   IF P THEN LPRINT "*************************";
490   IF P THEN LPRINT:LPRINT
500   R=T-1-INT((T-1)/7)*7
510   LE=21+LEN(DAY$(R))+LEN(M$(M))+LEN(STR$(D))+LEN(RIGHT$(STR$(Y),2))
520   V2=INT((64-LE)/2)
530   IF P THEN LPRINT SPC(2):FOR I=1 TO V2:LPRINT "*";:NEXT
540   IF P THEN LPRINT "STARTING DATE: ";DAY$(R);", ";M$(M);STR$(D);", 19";
550   PRINT RIGHT$(STR$(Y),2);:COLOR 7,0
560   IF P THEN LPRINT RIGHT$(STR$(Y),2);
570   IF P THEN LPRINT "****************";
580   IF P THEN LPRINT:LPRINT
590   LL=LEN(DAY$(RB))+LEN(M$(MB))+LEN(STR$(DB))+LEN(RIGHT$(STR$(YB),2))
600   LB=LL+18
610   VB=INT((64-LB)/2)
620   IF P THEN LPRINT SPC(2)
630   IF P THEN LPRINT "*************** BIRTH DATE: ";DAY$(RB);",
";M$(MB);STR$(DB);", 19";
640   IF P THEN LPRINT RIGHT$(STR$(YB),2);
650   IF P THEN LPRINT "****************"
660   IF P THEN LPRINT:LPRINT:LPRINT
670   IF P THEN LPRINT TAB(20);"DOWN",TAB(39);"CRITICAL",TAB(62);"UP"
680   IF P THEN LPRINT D$(R);D;M$(M);
690   F(0)=42:F$(0)="*"
700   F(1)=42+FNI(X):F$(1)="I"
710   F(2)=42+FNE(X):F$(2)="E"
720   F(3)=42+FNP(X):F$(3)="P"
730   FOR I=0 TO 2
740   FOR J=I+1 TO 3
750   IF F(I)<F(J) THEN 800
760   IF F(I)=F(J) THEN F(I)=0:F$(I)="":F$(J)="X":GOTO 800
770   Q=F(I):Q$=F$(I)
780   F(I)=F(J):F$(I)=F$(J)
790   F(J)=Q:F$(J)=Q$
800   NEXT J:NEXT I
810   FOR I=0 TO 3
820   IF P THEN LPRINT TAB(F(I));F$(I);
830   NEXT
840   X=X+.5:IF P THEN LPRINT
850   IF INT(X)<>X THEN 690
860   IF Z=S THEN 930
870   R=R+1:IF R=7 THEN R=0
880   D=D+1:IF M=4 OR M=6 OR M=9 OR M=11 THEN L=30:GOTO 920
890   IF M=2 AND Y/4=INT(Y/4) THEN L=29:GOTO 920
900   IF M=2 THEN L=28:GOTO 920
910   L=31
920   IF D>L THEN D=D-L:M=M+1:GOTO 940
930   IF P THEN LPRINT D$(R);D;:Z=Z+1:GOTO 690
940   IF M=13 THEN M=1:Z=Z+1:GOTO 660
950   Z=Z+1:IF D=1 THEN GOTO 660 ELSE GOTO 690
960   IF P THEN LPRINT:LPRINT
970   IF P THEN LPRINT CHR$(12)
980   INPUT "Do you want another Biorhythm";A$:IF LEFT$(A$,1)="Y" OR
LEFT$(A$,1)="y" THEN CLEAR 2000:P=0:GOTO 70
990   END
```

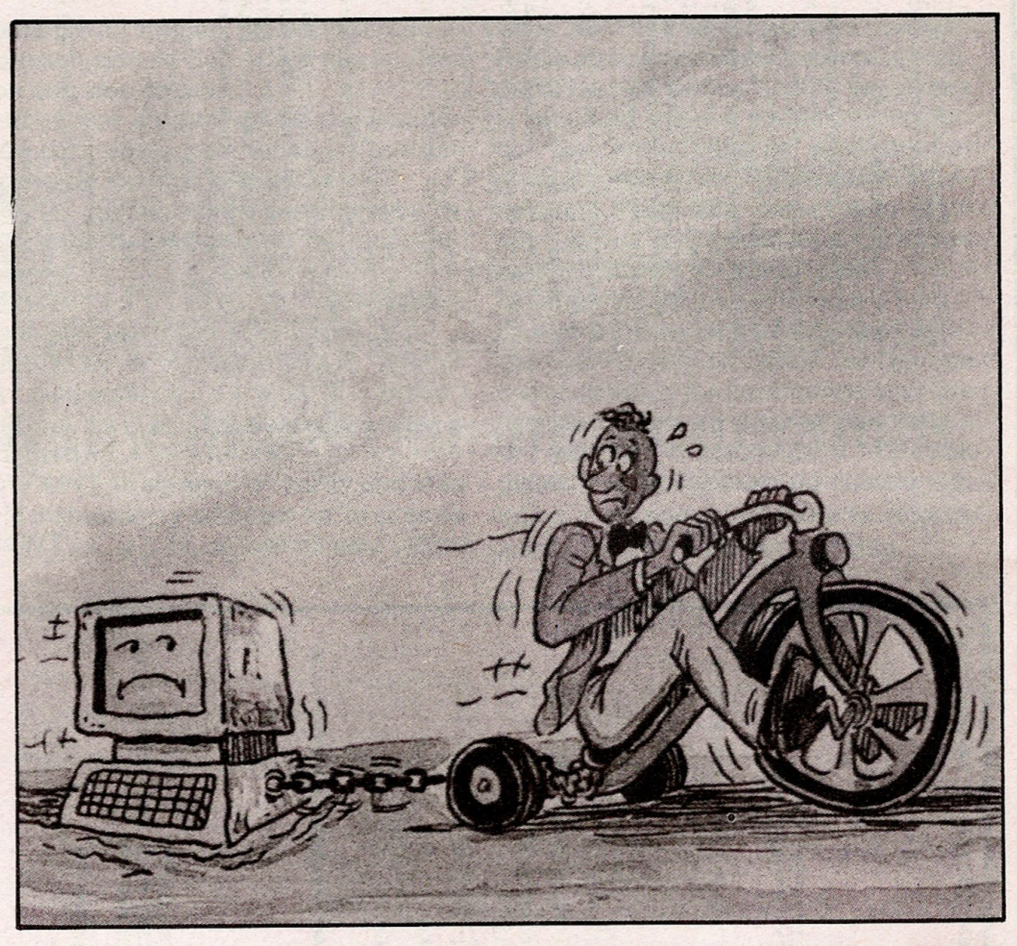
COMO ACELERAR UN PROGRAMA BASIC CON LA MSX

Retomamos la idea de acelerar la ejecución de los programas BASIC empleando algunos trucos de programación. Para los que recién nos sintonizan, recordemos algunos conceptos vertidos en la nota anterior.

Si somos asiduos usuarios del intérprete BASIC, tarde o temprano nos habremos hecho esta pregunta: ¿Por qué es lento mi programa?
Probablemente habremos pensado en inyectarle combustible de avión para acelerar el funcionamiento de su computadora.
La velocidad de ejecución de los programas es un tema clásico dentro de los usuarios de microcomputadoras (como la MSX). El proceso que consiste en la medición de velocidad de un programa o sistema versus otro equivalente se lleva a cabo mediante programas que se denominan "benchmarks" o pruebas de escritorio. En la jerga informática de nuestro país se denomina "prueba de escritorio" aquella que se realiza sobre lápiz y papel, así que dejaremos el término yanqui "benchmark". Este proceso involucra la ejecución con tomas de tiempo de dos programas similares funcionalmente y la verificación de las diferencias de tiempo de corrida. Es una herramienta valiosa para comparar a dos computadoras, pero no sirve para mucho más.

En este artículo discutiremos diversas reglas que la práctica en el uso de nuestro bienamado intérprete BASIC nos ha dictado para acelerar el funcionamiento de los programas. Para verificar la diferencias de tiempo de ejecución usaremos un programita sencillo que iremos modificando hasta lograr una reducción interesante.

Los tiempos y ejemplos se han tomado del intérprete MSX- BASIC y del MBASIC o BASIC 80 versión 5.21, que es la versión de BASIC que acompaña al compilador BASIC 80 disponible en plaza para nuestra norma. Sin embargo, aquel usuario que se tope con nuestras líneas podrá hacer uso de las reglas aquí brindadas ya que se pueden aplicar en general a cualquier intérprete BASIC.



En el artículo anterior descubrimos 6 "reglas sagradas", que reproducimos aquí:
REGLA 1: ELIMINE TODOS LOS COMENTARIOS (REM'S) DE LOS CICLOS DEL PROGRAMA
REGLA 2: SIEMPRE QUE SEA POSIBLE, UTILICE CICLOS FOR-NEXT EN VEZ DE CICLOS CONSTRUIDOS CON SENTENCIAS GOTO.
REGLA 3: LOS CICLOS FOR-NEXT SE EJECUTAN MAS RAPIDAMENTE CUANDO NO SE ESPECIFICA LA VARIABLE EN LA SENTENCIA NEXT
REGLA 4: SI EL MISMO CALCULO SE EFECTUA MAS DE UNA VEZ, EL TIEMPO DE EJECUCION PUEDE ABREVIARSE A VECES UTILIZANDO UNA VARIABLE PARA ALMACENAR EL RESULTADO Y LUEGO APLICARLA EN LA FORMULA, EN VEZ DE REPETIR EL CALCULO.
REGLA 5: SI SE UTILIZA MUCHAS VECES UNA CONSTANTE DENTRO DE UN PROGRAMA, ENTONCES ALMACENE SU VALOR EN UNA VARIABLE Y LUEGO USE LA VARIABLE EN VEZ DE LA CONSTANTE.
REGLA 6: SI LAS REGLAS ANTERIORES NO ACELERAN LO SUFICIENTE A SU PROGRAMA, TRATE CON OTRO ALGORITMO
Continuemos, pues, con nuestra investigación.
La siguiente regla que investigaremos es:
REGLA 7: REMPLACE LA EXPONENCIACION, COMO POR EJEMPLO R^2

Y R^3 POR LAS CORRESPONDIENTES MULTIPLICACIONES, R*R Y R*R*R.
Esta regla surge del hecho de que la función de exponente es una de las que más tiempo consume. Los programas que veremos a continuación muestran cuánto tiempo se puede llegar a ahorrar. Ambos programas calculan el área de un círculo con un radio de14 en un ciclo de 1000 repeticiones. La Tabla 1 indica la diferencia de velocidad así obtenida.

Antes de la regla 7

```
10 LET PI=3.14159
20 LET R=14
30 FOR N=1 TO 1000
40 LET AREA=PI*R^2
50 NEXT
```

Después de la regla 7

```
10 LET PI=3.14159
20 LET R=14
30 FOR N=1 TO 1000
40 LET AREA=PI*R*R
50 NEXT
```

TABLA 1: Aplicando la REGLA 7 (tiempo en segs).

	MSX	MBASIC
Antes	13.13	38.10
Después	10.40	7.53

REGLA 8: ELIMINE LOS PARENTESIS INNECESARIOS.

Cuando uno llega a captar cabalmente el tema de la prioridad algebraica, o sea, el orden de prioridades que utiliza el BASIC para resolver un conjunto de operaciones matemáticas, es sencillo eliminar los paréntesis que no se utilizan realmente. Por ejemplo, comparemos la velocidad de ejecución de los siguientes programas:

Antes de la regla 8:

```
10 LET A=2
20 LET B=3
30 FOR N=1 TO 1000
40 LET C=(A*B)-N
50 NEXT
```

Después de la regla 8:

```
10 LET A=2
20 LET B=3
30 FOR N=1 TO 1000
40 LET C=A*B-N
50 NEXT
```

TABLA 2: Aplicando REGLA 8 (tiempo en segs.).

	MSX	MBASIC
Antes	9.12	7.41
Después	8.53	7.10

En este caso (ver Tabla 2), se obtiene un 7 % de incremento en la velocidad, sabiendo únicamente que cuando no existen paréntesis, las multiplicaciones se calculan antes que las restas. El orden de prioridad del MSX BASIC es el siguiente:

1. Funciones (por ejemplo ABS, INT, etcétera.)
2. Potencias.
3. Signo negativo.
4. Productos y cocientes.
5. División entera.
6. Resto (MOD).
7. Sumas y restas.
8. Operaciones relacionales (<, >, =,etcétera.)
9. Operaciones lógicas (NOT, AND, OR, XOR, EQV, IMP).

La regla 9 muestra cómo usar las sentencias condicionales (IF-THEN) en forma más eficiente. Supongamos por ejemplo que deseamos verificar un número A, para ver si es positivo y luego imprimimos un mensaje "ad-hoc" indicando el resultado. Existen varias formas de implementar esta rutina:

SEGMENTO A:

```
10 IF A>0 THEN PRINT "A ES POSITIVO" ELSE PRINT "A NO ES POSITIVO"
```

SEGMENTO B:

```
10 IF A>0 THEN PRINT "A ES POSITIVO"
20 IF A<=0 THEN PRINT "A NO ES POSITIVO"
30
```

SEGMENTO C:

```
10 IF A>0 THEN 40
20 PRINT "A NO ES POSITIVO"
30 GOTO 50
40 PRINT "A ES POSITIVO"
50
```

SEGMENTO D:

```
10 IF A>0 THEN PRINT "A ES POSITIVO":GOTO 30
20 PRINT "A NO ES POSITIVO"
30
```

El segmento A es la mejor implementación, ya que es fácil de entender e interpretar, y además es la más veloz. El segmento B también es fácil de leer, pero es el más lento. El segmento C es más rápido, pero no se entiende tan fácilmente y el segmento D es una buena solución de compromiso entre legibilidad y velocidad.

Sin embargo, de lo visto anteriormente podemos plantear la REGLA "sagrada" 9:

REGLA 9: SIEMPRE QUE SE PUEDA, LA FORMA MAS EFICIENTE DE PLANTEAR UNA CONDICION ES "IF-THEN-ELSE".

La situación es más compleja cuando se deben aplicar más de dos condiciones posibles. Supongamos, por ejemplo, que deseamos optimizar la porción de programa que cuenta la cantidad de números positivos, de negativos y de ceros. Analicemos el siguiente segmento:

SEGMENTO E:

```
10 IF A=0 THEN CEROS=CEROS+1:GOTO 40
20 IF A<0 THEN NEG=NEG+1:GOTO 40
30 MAS=MAS+1
40
```

Ahora, si sabemos que la mayoría de los números serán positivos y existen muy pocos ceros, sería interesante reordenar el SEGMENTO F para obtener:

SEGMENTO F:

```
10 IF A>0 THEN MAS=MAS+1:GOTO 40
20 IF A<0 THEN NEG=NEG+1:GOTO 40
30 CEROS=CEROS+1
40
```

En el segmento F, se dedica muy poco tiempo a evaluar sentencias que tienen pocas oportunidades de cumplirse. En una ejecución real, el segmento F se ejecuta un 25 % más rápido que el segmento E. Llegamos así a la REGLA "sagrada" 10:

REGLA 10: ORDENE LAS SENTENCIAS "IF-THEN" DE TAL MANERA QUE LAS CONDICIONES QUE SE CUMPLAN MAS FACILMENTE SE VERIFIQUEN PRIMERO.

Muy relacionada con la REGLA 10, y lamentablemente con más posibilidades de que así ocurra, tenemos la siguiente regla:

REGLA 11: ORDENE LAS SENTENCIAS "IF-THEN" DE MANERA QUE LAS CONDICIONES QUE CONSUMAN MENOS TIEMPO SE VERIFIQUEN PRIMERO.

Veamos un ejemplo donde se aplica la regla 11. Supongamos que tenemos que probar un conjunto de alfanuméricos para ver si cumplen con dos condiciones, de tal manera que si no cumplen con alguna de las dos, el alfanumérico es considerado "ilegal". Las pruebas que deben pasar son:

PRUEBA A: El alfanumérico debe estar compuesto sólo por letras mayúsculas.

PRUEBA B: La longitud del alfanumérico debe ser menor que 9 caracteres.

Nótese que la PRUEBA A requiere la ejecución de un ciclo y la función MID$, mientras la PRUEBA B sólo requiere la ejecución de la rápida función LEN. A igualdad de condiciones, tiene más sentido aplicar primero la PRUEBA B. Si no se cumple la PRUEBA B, se evita la ejecución de la PRUEBA A, que tanto tiempo consume.

Para acercarnos a la siguiente regla, supongamos que ha estado realizando un ajuste fino de un programa y éste ha determinado que el costo mayor en tiempo de ejecución está relacionado con un determinado ciclo. Dentro de este ciclo, una variable es verificada un cierto número de veces. Una forma sencilla de acelerar la ejecución de este ciclo es inicializar simplemente la variable al comienzo del programa. Esta inicialización temprana permite que la variable se ubique al principio de la tabla simbólica de variables cuando se inicializa la misma al principio de la ejecución del programa. Cada vez que se solicita el valor de esta variable, la tabla simbólica se revisa, y las entradas que se ubican al principio se encuentran (obviamente) más rápido.

Como ejemplo de qué tan bien se mejora la performance, consideremos el siguiente segmento de programa, que obtiene un carácter de alguna fuente, digamos un archivo, y cuenta la cantidad de vocales y consonantes. En nuestro ejemplo, asumamos que se utilizan diez variables alfanuméricas en las líneas 10-90 (haciendo que se incorporen diez entradas a la tabla de variables). Agregando

```
5 LET A$="A"
```

logramos ubicar la variable A$ al principio de la tabla simbólica, y así recortamos sin más trámite 2 segundos de tiempo de ejecución del ciclo. Este es un buen lugar para recordar la REGLA 3 que sugiere que un NEXT reemplaza eficientemente dentro de los ciclos FOR a la sentencia NEXT N. Si se utiliza NEXT N, como la variable N está localizada al final de la tabla simbólica grande, la búsqueda de N puede hacer perder un tiempo considerable.

```
10 REM ASUMA QUE 10 VARIABLES SE INTRODUCEN EN LAS LINEAS 10-90
.
.
.
100 FOR N=1 TO 1000
110 REM ASUMA QUE A$ SE OBTUVO DE ALGUNA FORMA CON ESTA LINEA
120 IF A$="A" OR A$="E" OR A$="I" OR A$="O" OR A$="U" THEN VOCAL=VOCAL+1:GOTO 140
130 CNS=CNS+1
140 NEXT
```

Para redondear:

REGLA 12: LAS VARIABLES QUE SE REFERENCIAN MAS FRECUENTEMENTE DEBEN INICIALIZARSE AL PRINCIPIO DEL PROGRAMA.

Es inevitable que un programa largo requiera el uso de subrutinas. Si se utilizan las mismas líneas varias veces en un programa, tiene sentido incluir estas líneas en una subrutina. La siguiente regla tiene que ver con la ubicación física de las subrutinas:

REGLA 13: COLOQUE LAS SUBRUTINAS EN LAS LINEAS QUE SE BUSCAN PRIMERO.

La aplicación de esta regla varía de sistema en sistema. Cuando el intérprete MBASIC encuentra un comando tal como GOSUB 1000, las líneas de programas se buscan -comenzando desde la menor del programa. Si el programa debe revisar muchas líneas antes de encontrar el número de línea deseado, el tiempo de ejecución se incrementa bastante. Esto implica que cuando una subrutina llamada frecuentemente se ubica hacia el final de



un programa, puede requerir un tiempo extra bastante importante. Sin embargo, el MSX BASIC emplea otro estilo de llamado: cuando encuentra por primera vez el número de línea del GOSUB, realiza la búsqueda tal como se dijo. Pero una vez que lo encontró, cambia (internamente) el número de línea por su equivalente en posición de memoria, evitando toda pérdida de tiempo posterior.

La regla final que analizaremos también involucra a las subrutinas. Es obvio que la llamada a la subrutina involucra trabajo previo, por lo menos, dos saltos deben realizarse -uno a la subrutina y otro para volver al programa principal. Si deseamos sacrificar el ahorro de espacio, y quizás perder algo de legibilidad, entonces lo cual ponemos la subrutina dentro del programa principal, puede permitir la obtención de reducciones de tiempo de ejecución.

REGLA 14: SE PUEDE AHORRAR TIEMPO ELIMINANDO EL TRABAJO PREVIO DE UNA LLAMADA A SUBRUTINA Y REEMPLAZANDOLA DIRECTAMENTE EN EL PROGRAMA PRINCIPAL.

Un ejemplo sencillo de aplicación nos muestra los segmentos de programa que listamos a continuación. (Cada segmento verifica el primer carácter de A$ e incrementa un contador si el carácter es igual a Z.)

Antes de la REGLA 14:

```
10 FOR N=1 TO 1000
15 REM ASUMAMOS QUE A$ SE OBTIENE EN ESTA LINEA
20 GOSUB 50
30 NEXT
40 END
50 IF LEFT$(A$,1)="Z" THEN CTAZ=CTAZ+1
60 RETURN
```

Después de la REGLA 14:

```
10 FOR N=1 TO 1000
15 REM ASUMAMOS QUE A$ SE OBTIENE EN ESTA LINEA
20 IF LEFT$(A$,1)="Z" THEN CTAZ=CTAZ+1
30 NEXT
40 END
```

TABLA 3: Aplicando la REGLA 14 (tiempo en segs).

	MSX	MBASIC
Antes	12.03	11.84
Después	10.94	9.41

CONCLUSION.

A pesar de habernos referido al proceso de acelerar programas teniendo en cuenta los ya existentes, queda claro que estas mismas reglas permitirán encarar nuevos programas más eficientes. Un programa "bueno" normalmente tiene que balancear la eficiencia con la estructura, pero las ideas que vimos aquí no involucran intercambios del tipo espacio/tiempo o compromisos con estructuras buenas de programación.

Las reglas que hemos presentado no representan, ni mucho menos, los únicos métodos de reducción de tiempo de ejecución. Queda por ver consideraciones relacionadas con las sentencias de entrada/salida (I/O), por ejemplo. Sin embargo, hemos cubierto las reglas de eficiencia más comunes que son lo suficientemente amplias. Usted mismo podrá crear nuevas reglas de eficiencia leyendo cuidadosamente el manual de uso que acompaña a su computadora.

Hugo Caro

CERVEUX

GRAFICOS EN MSX2

Comenzamos a aprovechar las ventajas de la nueva computadora. En relación a su capacidad gráfica, sus cuatro modos gráficos extra para trabajar y la posibilidad de usar 80 columnas en SCREEN 0, explicamos como utilizarlos desde el BASIC.



Como todos sabemos, muy pronto (tal vez cuando lean esta nota) estará en la calle la Talent MSX 2. Y sabemos también que sus diferencias principales en cuanto a "la uno" están en su capacidad gráfica.
El chip de video V9938 de Yamaha nos ofrece cuatro modos gráficos extra para trabajar, además de permitirnos usar ochenta columnas en SCREEN 0.
¿Pero cómo utilizamos todo esto desde BASIC? Veamos de a poco:

LAS PANTALLAS

En SCREEN 0 basta con pedir WIDTH 80 para tener esa cantidad de caracteres por línea.
Siguen las SCREEN de 1 a 3 exactamente igual que antes, y continúan:
SCREEN 4, que es exactamente igual que SCREEN 2 pero con Sprites Avanzados(1).
SCREEN 5 nos eleva la resolución vertical a 212 (256 horizontal) y seguimos con 16 colores, pero a partir de aquí sea cual sea la cantidad de colores de un modo, estos son seleccionables entre 512.
SCREEN 6 duplica la resolución horizontal, y quedan 512 por 212 puntos, con cuatro colores.
SCREEN 7 mantiene la resolución de la anterior, pero con 16 colores.
SCREEN 8 vuelve a 256 por 212, pero con 256 colores.
Las pantallas 7 y 8 sólo están disponibles para computadoras con 128 K de RAM de Video, pero a no preocuparse, que la Talent la tiene.
Esto, además, nos permite manejar más de una pantalla a la vez; son las llamadas "páginas". En SCREEN 5 y 6, cuatro (de 0 a 3) y en SCREEN 7 y 8, dos (0 y 1).
Hay una capacidad especial del generador de video que se controla desde un nuevo parámetro agregado a la orden SCREEN. Recordemos que detrás del modo vienen el tamaño de los sprites, el click de las teclas, la velocidad de grabación de casete y la opción de impresora. En MSX 2 se agrega el modo de visualización, que puede ir de 0 a 3.
0 es el modo normal.
1 es el modo "entrelazado", en el que se ven dos páginas a la vez en pantalla, duplicándose la resolución vertical. Pero ojo, esto no quiere decir que yo pueda trabajar con el doble de altura, ya que las páginas siguen siendo independientes.
2 indica modo no entrelazado, con visualización alternativa de páginas pares e impares.
3 actúa igual, pero en entrelazado. En cualquiera de los dos últimos el número de página activada como visible debe ser impar.
Las distintas páginas son seleccionadas con el comando SET PAGE, al que le decimos primero cuál activamos como visible y segundo cuál como activa, de manera que podemos, por ejemplo, mostrar una pantalla mientras dibujamos en la otra.

LOS COLORES

La orden COLOR, ilustre habitante de la tecla F1, permanece prácticamente inalterada. Sigue indicando los colores para el frente, el fondo y los bordes. Cambia apenas la cantidad de colores que podemos utilizar, de acuerdo con el modo en el que trabajamos.
En SCREEN 6 hay colores de 0 a 3, en SCREEN 5 y 7 siguen entre 0 y 15, y en SCREEN 8 entre 0 y 255.
Sin embargo, dijimos que en todo momento podemos seleccionar esos colores de entre 512. ¿Cómo?
Aquí tenemos que ver la idea de RGB. Esta es la sigla que designa el sistema de colores generados a través de Rojo, Verde y Azul (Red, Green, Blue). Mezclándolos como si fueramos Da Vinci obtenemos los matices que deseamos. Esta alusión a Leonardo no es en vano, ya que según las proporciones en que mezclamos nuestros tres componentes obtenemos lo que

CERVEUX

se llama una "paleta".
Nuestra tablita de mezcla se llama COLOR= y es una función nueva que no debemos confundir con su predecesora COLOR. Con COLOR=(número de paleta, rojo, verde, azul) definimos qué color es realmente el que mencionamos en COLOR frente, fondo, borde. Las cantidades que podemos poner son de 0 a 7 para rojo, de 0 a 7 para verde y de 0 a 3 para azul. Algo interesante es que si variamos la mezcla de un color con el cual dibujamos algo en la pantalla, vemos la variación. Esto da lugar a efectos interesantes de brillo, etcétera.
Los colores de los sprites en las SCREEN de 4 a 8 pueden tener un color diferente por línea, pero hablar de ellos me llevaría otro artículo como este, así que lo dejo pendiente para un próximo número.

DIBUJANDO CON LOGICA

Para dibujar, los comandos siguen siendo los mismos salvando, claro, las distintas magnitudes que nos permite la resolución de cada modo. Pero además, en las instrucciones LINE, PSET y PRESET se nos agrega un parámetro extra: la función lógica que se efectúa entre el nuevo color del punto y el del punto viejo en pantalla, y que puede ser:
OR, que suma los dos colores.
AND, que los multiplica.
XOR, donde se suma la negación del viejo multiplicado por el nuevo más la negación del nuevo por el viejo.
PSET, que sustituye uno por el otro, y sería el modo normal.
PRESET niega el color nuevo. Cualquiera de estas funciones puede tener una "T" adelante (por ejemplo: TXOR), que implica que si el color nuevo es transparente, el color viejo no es modificado.

COPIONES GRAFICOS

Otro "chiche" de la MSX2 es el comando COPY en todas sus variantes.
Vayamos por partes:
Podemos copiar un área rectangular cualquiera de la pantalla en otra posición especificando: COPY (x1,y1)-(x2,y2) to (x3,y3) donde obviamente las coordenadas 1 y 2 son los vértices opuestos del rectángulo y las coordenadas 3 son el punto de destino. Pero la sintaxis completa incluye que, después de (x1,y1)-(x2,y2) podemos agregar el número de página a la que pertenece esa porción, y luego de TO (x3,y3), la página de destino y la operación lógica, de manera que puedo copiar de una página a la otra .
Otra opción es indicarle en vez de coordenadas de destino, el nombre de variable dimensionada o un nombre de archivo, o sea que puedo transferir a RAM, casete o disquete, incluyendo también cualquier combinación entre estos.

OTRAS POSIBILIDADES

Como ven hasta aquí, los gráficos crecieron bastante, pero uno de los puntos más interesantes y que nos abre más perspectivas en la norma 2 de MSX es la digitalización de imágenes, controlada por instrucciones sencillas como COPY SCREEN, SET VIDEO, etcétera. Tienen un trasfondo teórico más complejo lo cual que me obliga nuevamente a dejarlas para más tarde.
Fanáticos de los gráficos, me despido de ustedes después de contarles el "por qué" de MSX 2, y comprometiéndome con esas dos notas: Sprites y Digitalización.

Martín Salías.

PROBANDO EL SONIDO



Esta vez cambiamos un poco el tema de nuestros programitas. Dejamos de lado la parte "seria" de la cosa e investigamos un poco las posibilidades de sonido de la MSX. Muchas veces necesitamos para los juegos ruidos de disparos, pasos, golpes o música. En general la parte melódica se resuelve con algún amigo músico, pero la parte rítmica nos crea el problema de tener que "fabricar" el sonido de los parches. Este programita utiliza uno de los canales del PSG (Generador de Sonido Programable): el canal "A", habilitado por el registro 7 para generar ruido. El registro 8 con valor 16 habilita la envolvente en lugar de un volumen constante. El tipo de ruido se selecciona con el 6, y en el programa lo voy variando para probar diferentes golpes. Los registros 12 y 1 3 regulan el período de la envolvente, o sea cuánto tarda en llegar desde el máximo de volumen hasta el mínimo o viceversa.
La idea es que prueben sonidos, elijan los que más les gusten y le tomen la mano al "críptico" SOUND.

```
1 '-----------------------
2 '-----Simil Batería-----
3 '-----------------------
4 '-----Martín Salías-----
5 '-----------------------
6 '--Club de Usuarios MSX--
7 '-----------------------
10 RU=100:F=10
20 CLS
30 FOR Q=1 TO 1 STEP 0
40 LOCATE 0,2:PRINT"RUIDO:"
50 LOCATE 0,4:PRINT"FREQ:"
60 LOCATE 10,2:PRINT RU
70 LOCATE 10,4:PRINT F
80 LOCATE 9,2:INPUT RU
90 LOCATE 9,4:INPUT F
100 SOUND 6,RU
110 SOUND 7,247
120 SOUND 8,16
130 SOUND 12,F
140 SOUND 13,8
150 NEXT
```

3er CONCURSO

DE PROGRAMAS

auspiciado por TELEMATICA S.A. que proveerá los siguientes Premios:

PRIMER PREMIO

UN PERIFERICO

(a elección entre un monitor, una diskettera y una impresora).



UNA BECA

para trabajar en el Departamento de Investigación y Desarrollo de Telemática S.A.

SEGUNDO PREMIO

UN PERIFERICO

(a elección entre un monitor, una diskettera y una impresora).

ESPECIAL

Entre los programas recibidos, algunos de ellos podrán ser editados por SYSTEMAC S.A., reconociéndose los derechos de autor

En caso de que el ganador no pueda utilizar la beca,será ofrecida a quien obtenga el segundo premio, y si éste tampoco pudiera aprovecharla se otorgará a alguno de los participantes del certamen que se hubiera destacado.

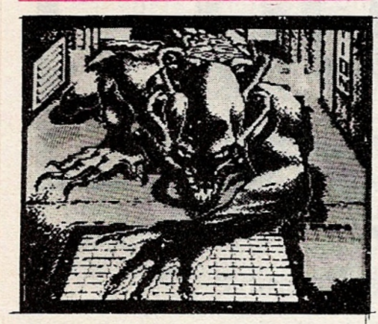
Se premiará el mejor software de cualquier clase (juegos, utilitarios, científico o comercial).

BASES: **No sólo será indispensable que el programa enviado en caset ó disket funcione correctamente, sino que además debe cumplir con ciertas reglas:**

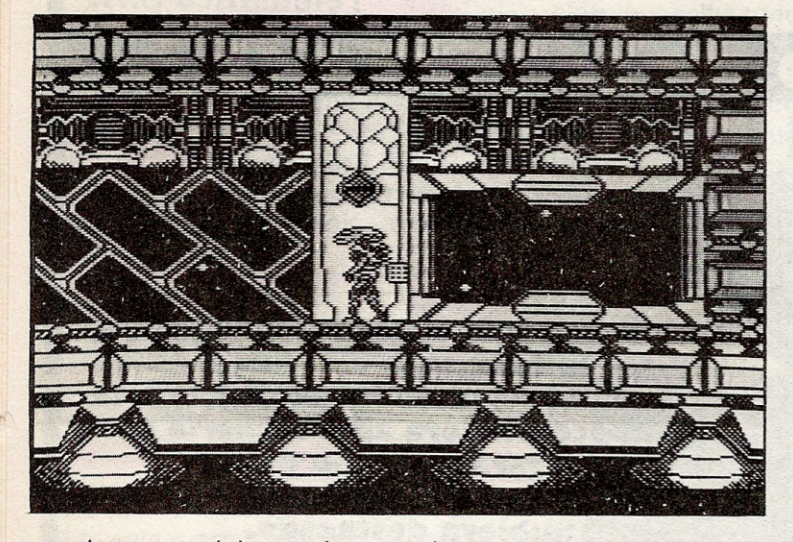
- Programación estructurada en bloques fácilmente diferenciables.
- Fácil seguimiento del mismo y detalle de éste como parte de su documentación. (Diagrama de bloques con los números de línea que los identifiquen)
- Aclaración y clara explicación de los algoritmos utilizados, deben figurar como parte de la documentación.
- Las variables y/o direcciones de memoria utilizados también se deben incluir en esta documentación.
- Listado de nemónicos assembler y la localización en memoria si es que se utiliza este tipo de lenguaje.
- Calidad y originalidad de gráficos, sonidos y pantallas de menú.

Los trabajos deberán enviarse antes del 30 de noviembre próximo (cierre del certámen) a: Paraná 720, piso 5º, (1017) Capital Federal.

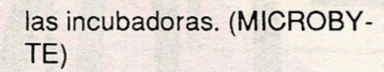
SURVIVOR



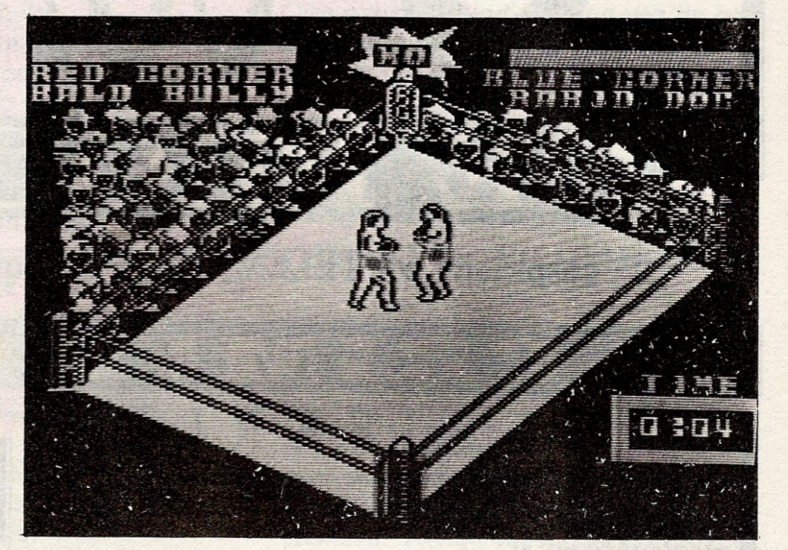
En algún lugar del espacio, girando en torno a un viejo planeta casi muerto, flota una inmensa astronave cargada con todas las formas de vida de un millar de mundos. La nave es oscura, extraña y poderosa, y dispone de energías capaces de desgarrar la



propia textura del espacio.
Sus amos son dueños de secretos tecnológicos que están más allá de toda imaginación, y sus palabras son ley para miles de razas en todas las galaxias. Y sin embargo, estos seres tan poderosos ¡tienen un terrible problema...!
En algún lugar de la enorme estructura, arrastrándose sinuosamente por cierto conducto de ventilación, o recorriendo los invernaderos y salas de cohtrol, o deslizándose por los transportadores, una solitaria criatura lucha por su supervivencia y por la de su raza.
La criatura es el producto final de un millón de años de evolución sobre un mundo agonizante. Astuta y llena de instintos salvajes, con músculos resistentes como el metal y ojos duros como el diamante, se resiste a los altivos dominadores.
El objetivo del juego es evitar la extinción de las razas introduciendo diez vainas en las incubadoras repartidas por toda la nave. El alienígena dispone de una energía que se irá agotando poco a poco. La única forma de reponer energía es soltando las vainas en las incubadoras, o bien devorando a los nanoingenieros que hay por la nave. La criatura puede defenderse escupiendo ácido paralizante o escondiéndose en conductos de ventilación.
La nave consta de 142 habitáculos repartidos en cuatro planos que se unen a través de las puertas o de rejillas de ventilación. Cada plano tiene varios niveles a los que se accede por medio de ascensores. Para entrar en un ascensor, una puerta o una rejilla, bastará con colocarse frente a ella y agacharse.
La criatura puede aumentar la potencia de sus saltos agachándose para tomar impulso. También puede lanzar el ácido a mayor distancia si se mantiene pulsada la tecla de disparo por más tiempo.
Podemos encuadrar a este entretenimiento dentro de los juegos de aventuras en los que hay que ir recogiendo cosas. Sólo que en este caso hay que dejar las vainas en las incubadoras. (MICROBYTE)



3D KNOCK OUT



Tenemos aquí un juego que nos enfrenta en un ring ante temibles y poderosos boxeadores. Sus nombres son Mad Joe, Matt Black, Hammerhead, Doc Martin, Wyatt Thug, Jo Violent, Eric T. Red y Big Jin. Cada uno nos enfrenta a un nivel de dificultad diferente y para vencerlos tendremos que poner toda nuestra sabiduría boxística (y el poder de nuestros puños). Las peleas son de 8 rounds de 2 minutos cada uno. Al fin de cada asalto la computadora dará la tarjeta del jurado. Gana quien haya sumado más asaltos a su favor o puesto fuera de combate al adversario.
Se puede jugar entre dos jugadores, o bien contra la máquina. El joystick da una capacidad de movimiento de 360 grados alrededor del ring. De esta manera se simula el efecto de la pelea y uno puede ir caminando en el ring, o bailando como Mohamed Alí, alrededor del adversario.
Con el botón apretado se puede lanzar una amplia gama de golpes al adversario y controlar la defensa.
En la parte superior de la pantalla se muestra la cantidad de energía de ambos contendientes, los golpes que se han dado y otras señales. Cada combatiente tiene diez de ellas al principio, y las va perdiendo a medida que recibe golpes. La energía y las señales se recuperan con el paso del tiempo.
Cuando se consigue batir a un oponente se pasará automáticamente a un nivel superior de habilidad. Exite un factor de atontamiento, que aumenta la habilidad y que, si está activado, tiene en cuenta los daños que producen los golpes a la cabeza. ¡Se apagan las luces y comienza el combate...! (MICROBYTE)

LA ABEJITA INQUIETA 1



La serie APRENDER JUGANDO de SYSTEMAC, de la que forma parte este programa, no pretende ser un sistema autosuficiente de aprendizaje preescolar, sino un instrumento más dentro de la gama de actividades que el niño debe realizar en esa etapa. Entre ellas está el movimiento corporal, la manipulación directa de objetos y la ex-

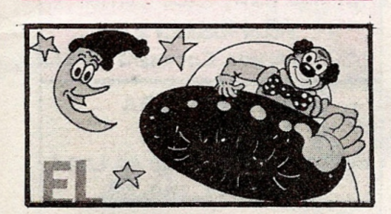
presión.

En LA ABEJITA INQUIETA 1, el objetivo es que el niño reconozca las formas geométricas que componen una figura dada, a fin de desarollar su capacidad de análisis y de síntesis, requisitos imprescindibles para el aprendizaje posterior de la lecto-escritura y el cálculo.

Consta de 5 figuras diferentes, compuestas por distintos elementos gráficos, entre los cuales el niño debe reconocer la forma geométrica que se está trabajando. Cuando se complete con éxito uno de los dibujos presentados, y antes de comenzar con el siguiente, el niño tendrá acceso a un juego recreativo, que consiste en el recorrido de un laberinto.

En cada laberinto se dispone de dos oportunidades. El manejo puede hacerse tanto con las teclas del cursor como con joystick. (SYSTEMAC)

EL CAZADOR DEL ESPACIO 1



El objetivo de este juego es que el niño diferencie objetos en función de su tamaño, independientemente de la forma o color que tengan. Al alcanzar este objetivo, el niño se prepara para la realización de clasificaciones de objetos en función de una sola característica.

Asimismo, este juego contribuye a mejorar su habilidad manual y su sentido de la orientación en el espacio, pues en él hay que desplazar la nave o la pantalla en todas las direcciones y además encajar, mediante el mando del juego, cada objeto con precisión dentro del recuadro del mismo tamaño.

EL CAZADOR ESPACIAL 1 consiste en dos juegos que muestran al niño un conjunto de objetos de diferentes tamaños que él tiene que ubicar en el recuadro que corresponda, según su tamaño.

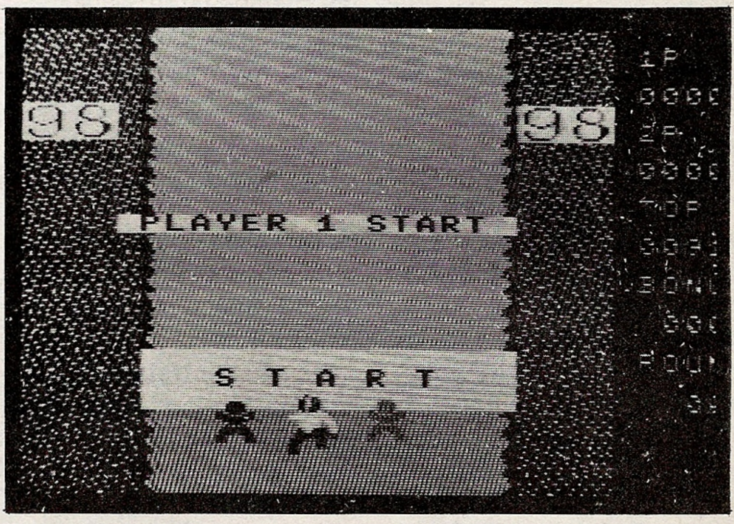
El juego 1 es un conjunto de 16 objetos grandes y pequeños, mientras que en el 2 los objetos son 125 y también hay objetos medianos. (SYSTEMAC)

LA ALFOMBRA MAGICA 2



Este juego presenta al niño el desafío de componer, elemento por elemento, la cantidad de objetos expresada por un número dado. También el proceso inverso, de familiarizarse con la descomposición del número, y poder contar progresivamente, del uno al nueve.

La realización del entretenimiento consiste en desplazar por la pantalla LA ALFOMBRA MAGICA y ordenar que Aladino corte con su sable, una por una, tantas manzanas de un árbol como indica la cifra que aparece sobre el lomo del camello. Una vez cortada



la cantidad correcta de manzanas, deber transportarlas hasta el camello para depositarlas en su joroba. Entre ejercicio y ejercicio hay un juego recreativo en el que el niño debe acertar, mediante el lanzamiento de bolillas, a los objetos que se desplazan con mayor o menor velocidad según la destreza manifestada por el participante. (SYSTEMAC)

TOP ROLLER

Aquí nos debemos enfrentar con una pista de patinaje para atravesarla deslizándonos sobre el hielo.

Claro que la tarea se ve dificultada (¿si no qué gracia tendría el juego?) por dos competidores que tratarán de sacarnos de la pista por todos los medios (golpeándonos, obstruyendo el camino, etcétera).

Con ellos no hay que tener misericordia. Debemos enfrentarlos con sus propias armas y sacarlos de la pista; pero esto será temporal, ya que volverán inmediatamente a la carga.

Otras dificultades se presentan por los obstáculos que hay en la pista: una aplanadora, unos cohetes, bombas que estallan y sectores en los que hay que saltar para superar el cerco.

Los paisajes van variando a medida que avanzamos y, obviamente, la dificultad se va incrementando.

La melodía que acompaña el desarrollo del juego es agradable y pegadiza. (VALENTE)

Para comunicarse con nosotros deben escribirnos a "Load Revista para usuarios de MSX", Paraná 720, 5º Piso, (1017), Capital Federal.

EL ARTE DE COPIAR

En el número 13 de la revista se publicó un programa para copiar de casete a disco. Al listado número 1 (en BASIC) lo grabé sin inconvenientes, en cambio con el de código máquina tuve problemas ya que no pude grabarlo.
Intenté hacerlo colocando números de línea delante de cada POKE pero luego no pude grabarlo en el disco.
Además me gustaría saber cómo se hace para "pokear" el listado de la figura 2.

HORACIO E. MELNYCZAJKO
CAPITAL FEDERAL

Load MSX

Vamos a tratar de aclarar el tema: la forma de grabar los datos de la figura 1 es introducirlos en primer lugar en la memoria de la máquina. Eso lo hacemos con la sentencia POKE, que da el número de posición de la memoria a modificar y el valor a introducir allí. Esos valores están dados en el sistema de numeración hexadecimal para la figura 1 y en decimal para la 2. La forma de introducirlos es, por ejemplo, POKE &HD000, &HCD para el primer caso y POKE 32768, 237 para el segundo. Luego se presiona return y ya está modificada esa memoria. Luego lo que hay que hacer es grabar esa memoria modificada en el disco BSAVE "DISK.ASM", &HD000, &HD016. La figura 2 se graba en casete mediante BSAVE "CAS:AFANO", 32768, 34303, 32768.
Con referencia a esta última hay que tener en cuenta que los diferentes números que hay a la derecha de las direcciones de memoria son los de las memorias sucesivas. Ejemplo: POKE 32768, 237: POKE 32769, 115: POKE 32770, 201, etcétera.

CONSULTA

Hace aproximadamente 60 días adquirí una computadora y una disquetera Talent.
Pese a haber tratado de leer toda la bibliografía con la que haber contado y consultar a mucha gente no he podido aclarar algunas cosas, y por lo tanto ruego si es posible tengan a bien explicármelas. Anteriormente yo contaba con una computadora SINCLAIR 2068. En la misma, cuando yo programaba utilizaba la sentencia DIM como en el siguiente ejemplo: DIM A$(10,20) dimensionaba 10 renglones de 20 elementos cada uno.
Leyendo y consultando veo que no es lo mismo en las MSX. Las preguntas son:
1.¿Es lo mismo DIM A$(5,2) que DIM A$(10)?
2.En un programa,luego de darle DIM A$(5,2), traté de ingresar datos de 20 caracteres a cada una de las posiciones de la matriz. Esto lo hizo sin problemas, pero cuando quise ingresar datos de 30 caracteres me dio OUT OF STRING SPACE. ¿Cómo puedo solucionarlo?

Sergio D. Guida
Argüello - Córdoba

Load MSX

1. No es lo mismo. Si bien la cantidad de información que podés almacenar es similar en ambos casos, la diferencia radica en que son matrices de dimensiones diferentes.
2. La memoria reservada para variables alfanuméricas tiene un tamaño normal de 200 bytes.
Cuando colocás 10 cadenas de 20 caracteres, estás en el límite de esa memoria. Obviamente, cuando se intenta colocar cadenas más largas sale el error que mencionás.
La solución es utilizar el comando CLEAR antes de dimensionar la matriz. En tu caso podrías colocar CLEAR 300

ESPERA

Cuando cargo un programa en mi computadora, una TALENT MSX, para cambiar luego de programa tengo que mantener apagada la máquina unos cinco segundos aproximadamente. De lo contrario queda un poco del viejo programa en la memoria. Mi pregunta es si esto que me pasa es normal.

ALEJANDRO M. ULANOSKY
Córdoba

Load MSX

Lo que sucede es normal. Cuando se ejecuta el programa se modifican muchas posiciones de memoria, que hay que "volver a la normalidad" cuando se "resetea" la máquina. Por este motivo es que los fabricantes de todas las marcas recomiendan dejar pasar un tiempo prudencial entre el apagado y reencendido de la computadora.

STAFF



Director General
Ernesto del Castillo

Director Editorial
Cristian Pusso

Director Periodístico
Fernando Flores

Secretario de Redacción
Ariel Testori

Arte y Diagramación
Fernando Amengual y Tamara Migelson

Departamento de Avisos
Oscar Devoto y Nelso Capello

Departamento de Publicidad
Dolores Urien

Servicios Fotográficos
Image Bank, Oscar Burriel, Víctor Grubicy y Eduardo Comesaña

FOTO DE TAPA
GLAURUNG
Software comentado en Load MSX Nº 11, pág. 32.

Load Revista para usuarios de la norma MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5º Piso, (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Radiollamada: Tel.: 311-0056 y 312-6383, código 5941. Registro Nacional de la Propiedad Intelectual: E. T. M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de la Propiedad Intelectual.

ISSN 0326-8241

Impresión: Calcotam, Fotocromo tapa: Columbia. Fotocomposición: Interamericana Gráfica.
Los ejemplares atrasados se venderán al precio del último número en circulación.

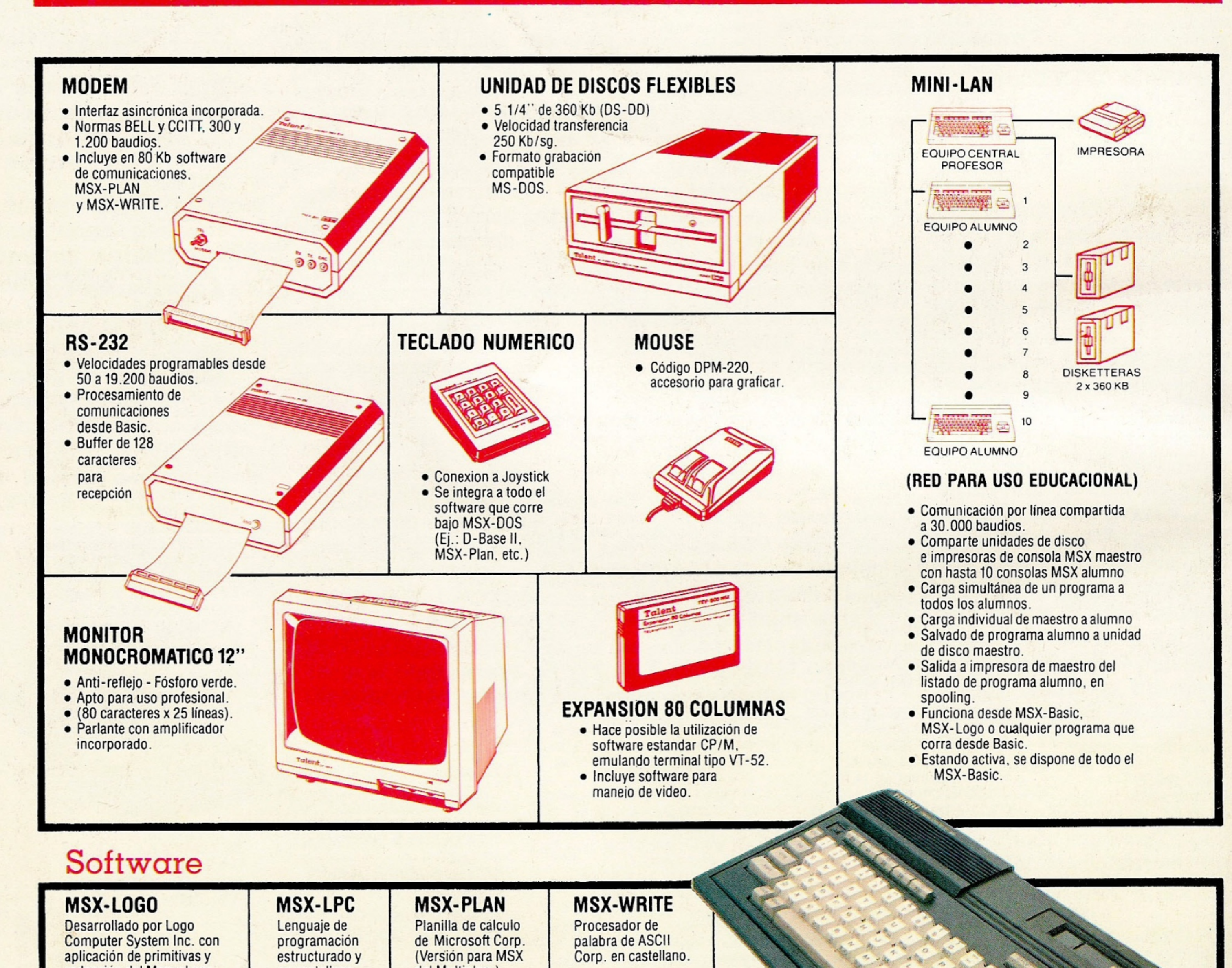
Distribuidor en Capital: Martino, Juan de Garay 358, P. B. Capital.
Distribuidor interior: D G P: Hipólito Yrigoyen 1450, Capital Federal. T.E. 38-9266/9800.





COMPUTACION
K64
PARA TODOS
NOVIEMBRE 1987
INEDITOS
APLICACIONES
DC-64/128, SPECTRUM, MSX, TI
MAQUINAS QUE
PIENSAN
GOLF
INFORMATICO
PC
SOFTWARE
AÑO 3 N° 32 # 7 REP. ARGENTINA
HARD y SOFT PARA ATARI, SPECTRUM, CZ, TK, DREAN COMMODORE, TI y MSX
Talent DM-120M
MSX2 TPC-310

Encienda una computadora Talent MSX y sus periféricos.



Software

MSX-LOGO	MSX-LPC	MSX-PLAN	MSX-WRITE
Desarrollado por Logo Computer System Inc. con aplicación de primitivas y redacción del Manual por los Ings. Hilario Fernández Long y Horacio Reggini.	Lenguaje de programación estructurado y en castellano.	Planilla de calculo de Microsoft Corp. (Versión para MSX del Multiplan.)	Procesador de palabra de ASCII Corp. en castellano.

Talent

Tecnología y Talento *en su casa*

Producida en San Luis por Telemática S.A. licenciataria exclusiva de Microsoft Corp. y ASCII Corp. para uso de la norma MSX en Argentina.
6 meses de garantía y mensualmente en su quiosco la revista Load MSX.